15

Matrices

In this chapter, we discuss basic definitions and results concerning matrices. We shall start out with a very general point of view, discussing matrices whose entries lie in an arbitrary ring R. Then we shall specialize to the case where the entries lie in a field F, where much more can be said.

One of the main goals of this chapter is to discuss "Gaussian elimination," which is an algorithm that allows us to efficiently compute bases for the image and kernel of an F-linear map.

In discussing the complexity of algorithms for matrices over a ring R, we shall treat a ring R as an "abstract data type," so that the running times of algorithms will be stated in terms of the number of arithmetic operations in R. If R is a finite ring, such as $\mathbb{Z}_m$, we can immediately translate this into a running time on a RAM (in later chapters, we will discuss other finite rings and efficient algorithms for doing arithmetic in them).

If R is, say, the field of rational numbers, a complete running time analysis would require an additional analysis of the sizes of the numbers that appear in the execution of the algorithm. We shall not attempt such an analysis here—however, we note that all the algorithms discussed in this chapter do in fact run in polynomial time when $R = \mathbb{Q}$, assuming we represent rational numbers as fractions in lowest terms. Another possible approach for dealing with rational numbers is to use floating point approximations. While this approach eliminates the size problem, it creates many new problems because of round-off errors. We shall not address any of these issues here.

15.1 Basic definitions and properties

Throughout this section, R denotes a ring.

For positive integers m and n, an $m \times n$ **matrix** A over a ring R is a

rectangular array

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix},$$

where each entry a_{ij} in the array is an element of R; the element a_{ij} is called the (i, j) **entry** of A, which we may denote by $A(i, j)$. For $i = 1, \ldots, m$, the i**th row** of A is

$$(a_{i1}, \ldots, a_{in}),$$

which we may denote by $A(i)$, and for $j = 1, \ldots, n$, the j**th column** of A is

$$\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix},$$

which we may denote by $A(\cdot, j)$. We regard a row of A as a $1 \times n$ matrix, and a column of A as an $m \times 1$ matrix.

The set of all $m \times n$ matrices over R is denoted by $R^{m \times n}$. Elements of $R^{1 \times n}$ are called **row vectors (of dimension** n**)** and elements of $R^{m \times 1}$ are called **column vectors (of dimension** m**)**. Elements of $R^{n \times n}$ are called **square matrices (of dimension** n**)**. We do not make a distinction between $R^{1 \times n}$ and $R^{\times n}$; that is, we view standard n-tuples as row vectors. Also, where there can be no confusion, we may interpret an element of $R^{1 \times 1}$ simply as an element of R.

We can define the familiar operations of scalar multiplication, addition, and multiplication on matrices:

- If $A \in R^{m \times n}$ and $c \in R$, then cA is the $m \times n$ matrix whose (i, j) entry is $cA(i, j)$.
- If $A, B \in R^{m \times n}$, then $A + B$ is the $m \times n$ matrix whose (i, j) entry is $A(i, j) + B(i, j)$.
- If $A \in R^{m \times n}$ and $B \in R^{n \times p}$, then AB is the $m \times p$ matrix whose (i, k) entry is

$$\sum_{j=1}^{n} A(i, j)B(j, k).$$

We can also define the difference $A - B := A + (-1_R)B$ of matrices of the same dimension, which is the same as taking the difference of corresponding entries. These operations satisfy the usual properties:

Theorem 15.1. *If $A, B, C \in R^{m\times n}$, $U, V \in R^{n\times p}$, $Z \in R^{p\times q}$, and $c, d \in R$, then*

(i) $c(dA) = (cd)A = d(cA)$,
(ii) $(A + B) + C = A + (B + C)$,
(iii) $A + B = B + A$,
(iv) $c(A + B) = cA + cB$,
(v) $(c + d)A = cA + dA$,
(vi) $(A + B)U = AU + BU$,
(vii) $A(U + V) = AU + AV$,
(viii) $c(AU) = (cA)U = A(cU)$,
(ix) $A(UZ) = (AU)Z$.

Proof. All of these are trivial, except the last one which requires just a bit of computation to show that the (i, ℓ) entry of both $A(UZ)$ and $(AU)Z$ is (verify)

$$\sum_{j=1}^{n}\sum_{k=1}^{p} A(i, j)U(j, k)Z(k, \ell). \quad \square$$

Note that while matrix addition is commutative, matrix multiplication in general is not.

Some simple but useful facts to keep in mind are the following:

- If $A \in R^{m\times n}$ and $B \in R^{n\times p}$, then the kth column of AB is equal to Av, where $v = B(\cdot, k)$; also, the ith row of AB is equal to wB, where $w = A(i)$.
- If $A \in R^{m\times n}$ and $u = (u_1, \ldots, u_m) \in R^{1\times m}$, then

  $$uA = \sum_{i=1}^{m} u_i A(i).$$

  In words: uA is a linear combination of the rows of A, with coefficients taken from the corresponding entries of u.
- If $A \in R^{m\times n}$ and

  $$v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \in R^{n\times 1},$$

then

$$Av = \sum_{j=1}^{n} v_j A(\cdot, j).$$

In words: Av is a linear combination of the columns of A, with coefficients taken from the corresponding entries of v.

If $A \in R^{m \times n}$, the **transpose** of A, denoted by $A^\top$, is defined to be the $n \times m$ matrix whose (j, i) entry is $A(i, j)$.

Theorem 15.2. *If $A, B \in R^{m \times n}$, $C \in R^{n \times p}$, and $c \in R$, then*

(i) $(A + B)^\top = A^\top + B^\top$,

(ii) $(cA)^\top = cA^\top$,

(iii) $(A^\top)^\top = A$,

(iv) $(AC)^\top = C^\top A^\top$.

Proof. Exercise. □

An $n \times n$ matrix A is called a **diagonal matrix** if $A(i, j) = 0_R$ for $i \neq j$, which is to say that the entries off the "main diagonal" of A are all zero. A **scalar matrix** is a diagonal matrix whose diagonal entries are all the same. The scalar matrix I, where all the entries on the main diagonal are 1_R, is called the $n \times n$ **identity matrix**. It is easy to see that if A is an $n \times n$ matrix, then $AI = IA = A$. More generally, if B is an $n \times m$ matrix, then $IB = B$, and if C is an $m \times n$ matrix, then $CI = C$.

If A_i is an $n_i \times n_{i+1}$ matrix, for $i = 1, \ldots, k$, then by associativity of matrix multiplication (part (ix) of Theorem 15.1), we may write the product matrix $A_1 \cdots A_k$, which is an $n_1 \times n_{k+1}$ matrix, without any ambiguity. For an $n \times n$ matrix A, and a positive integer k, we write A^k to denote the product $A \cdots A$, where there are k terms in the product. Note that $A^1 = A$. We may extend this notation to $k = 0$, defining A^0 to be the $n \times n$ identity matrix.

One may readily verify the usual rules of exponent arithmetic: for non-negative integers k_1, k_2, we have

$$(A^{k_1})^{k_2} = A^{k_1 k_2} \quad \text{and} \quad A^{k_1} A^{k_2} = A^{k_1 + k_2}.$$

It is easy also to see that part (iv) of Theorem 15.2 implies that for all non-negative integers k, we have

$$(A^k)^\top = (A^\top)^k.$$

Algorithmic issues

For computational purposes, matrices are represented in the obvious way as arrays of elements of R. As remarked at the beginning of this chapter, we shall treat R as an "abstract data type," and not worry about how elements of R are actually represented; in discussing the complexity of algorithms, we shall simply count "operations in R," by which we mean additions, subtractions, multiplications; we shall sometimes also include equality testing and computing multiplicative inverses as "operations in R." In any real implementation, there will be other costs, such as incrementing counters, and so on, which we may safely ignore, as long as their number is at most proportional to the number of operations in R.

The following statements are easy to verify:

- We can multiply an $m \times n$ matrix times a scalar using mn operations in R.
- We can add two $m \times n$ matrices using mn operations in R.
- We can multiply an $m \times n$ matrix and an $n \times p$ matrix using $O(mnp)$ operations in R.

It is also easy to see that given an $m \times m$ matrix A, and a non-negative integer e, we can adapt the repeated squaring algorithm discussed in §3.4 so as to compute A^e using $O(\mathrm{len}(e))$ multiplications of $m \times m$ matrices, and hence $O(\mathrm{len}(e)m^3)$ operations in R.

15.2 Matrices and linear maps

Let R be a ring.

For positive integers m and n, we may naturally view $R^{1\times m}$ and $R^{1\times n}$ as R-modules. If A is an $m \times n$ matrix over R, then the map σ that sends $v \in R^{1\times m}$ to $vA \in R^{1\times n}$ is easily seen to be an R-linear map. Evidently, σ is injective if and only if the rows of A are linearly independent, and σ is surjective if and only if the rows of A span $R^{1\times n}$. Likewise, the map τ that sends $w \in R^{n\times 1}$ to $Aw \in R^{m\times 1}$ is also an R-linear map. Again, τ is injective if and only if the columns of A are linearly independent, and τ is surjective if and only if the columns of A span $R^{m\times 1}$.

Thus, the matrix A defines in a natural way two different linear maps, one defined in terms of multiplying a row vector on the right by A, and the other in terms multiplying a column vector on the left by A. With either of these interpretations as a linear map, matrix multiplication has a natural interpretation as function composition. Let $A \in R^{m\times n}$ and $B \in R^{n\times p}$, and consider the product matrix $C = AB$. Let $\sigma_A, \sigma_B, \sigma_C$ be the maps defined

by multiplication on the right by A, B, C, and let τ_A, τ_B, τ_C be the maps defined by multiplication on the left by A, B, C. Then it easily follows from the associativity of matrix multiplication that $\sigma_C = \sigma_B \circ \sigma_A$ and $\tau_C = \tau_A \circ \tau_B$.

We have seen how matrix/vector multiplication defines a linear map. Conversely, we shall now see that the action of any R-linear map can be viewed as a matrix/vector multiplication, provided the R-modules involved have bases (which will always be the case for finite dimensional vector spaces).

Let M be an R-module, and suppose that $\mathcal{A} = (\alpha_1, \ldots, \alpha_m)$, with $m > 0$, is a basis for M. In this setting, the ordering of the basis elements is important, and so we refer to $\mathcal{A}$ as an **ordered basis**. Now, $\mathcal{A}$ defines a canonical R-module isomorphism ϵ that sends $(a_1, \ldots, a_m) \in R^{1\times m}$ to $a_1\alpha_1 + \cdots + a_m\alpha_m \in M$. Thus, elements of M can be represented concretely as elements of $R^{1\times m}$; however, this representation depends on the choice $\mathcal{A}$ of the ordered basis. The vector $\epsilon^{-1}(\alpha)$ is called the **coordinate vector of α (with respect to $\mathcal{A}$)**.

Let N be an R-module, and suppose $\mathcal{B} = (\beta_1, \ldots, \beta_n)$, with $n > 0$, is an ordered basis for N. Just as in the previous paragraph, $\mathcal{B}$ defines a canonical R-module isomorphism $\delta : R^{1\times n} \to N$.

Now let $\rho : M \to N$ be an arbitrary R-linear map. For any $\alpha \in M$, if $\alpha = \epsilon(a_1, \ldots, a_m)$, then because ρ is R-linear, we have

$$\rho(\alpha) = \sum_{i=1}^{m} \rho(a_i\alpha_i) = \sum_{i=1}^{m} a_i\rho(\alpha_i).$$

Thus, the action of ρ on M is completely determined by its action on the α_i.

Let us now define an $m \times n$ matrix T whose ith row, for $i = 1, \ldots, m$, is defined to be $\delta^{-1}(\rho(\alpha_i))$, that is, the coordinate vector of $\rho(\alpha_i)$ with respect to the ordered basis $\mathcal{B}$. With T defined in this way, then for any $\alpha \in M$ we have

$$\delta^{-1}(\rho(\alpha)) = \epsilon^{-1}(\alpha)T.$$

In words: if we multiply the coordinate vector of α on the right by T, we get the coordinate vector of $\rho(\alpha)$.

A special case of the above is when $M = R^{1\times m}$ and $N = R^{1\times n}$, and $\mathcal{A}$ and $\mathcal{B}$ are the standard bases for M and N (i.e., for $i = 1, \ldots, m$, the ith vector of $\mathcal{A}$ has a 1 in position i and is zero everywhere else, and similarly for $\mathcal{B}$). In this case, $\rho(v) = vT$ for all $v \in R^{1\times m}$.

To summarize, we see that an R-linear map ρ from M to N, together with particular ordered bases for M and N, uniquely determine a matrix T such

that the action of multiplication on the right by T implements the action of ρ with respect to the given ordered bases. There may be many ordered bases for M and N to choose from, and different choices will in general lead to different matrices. In any case, from a computational perspective, the matrix T gives us an efficient way to compute the map ρ, assuming elements of M and N are represented as coordinate vectors with respect to the given ordered bases.

Of course, if one prefers, by simply transposing everything, one can equally well represent the action of ρ in terms of the action of multiplication of a column vector on the left by a matrix.

***Example* 15.1.** Consider again the ring $E = R[\mathtt{X}]/(f)$, where $f \in R[\mathtt{X}]$ is monic of degree ℓ, and suppose that $\ell > 0$ (see Examples 9.34, 9.43, 14.3, and 14.22). Let $f = f_0 + f_1\mathtt{X} + \cdots f_{\ell-1}\mathtt{X}^{\ell-1} + \mathtt{X}^{\ell}$, where $f_0, \ldots, f_{\ell-1} \in R$. Consider the element $\eta = [\mathtt{X}]_f \in E$. Let $\rho : E \to E$ be the η-multiplication map, that is, the map that sends $\alpha \in E$ to $\eta\alpha \in E$. This is an R-linear map, and the matrix $T \in R^{\ell \times \ell}$ that represents this map with respect to the ordered basis $1, \eta, \eta^2, \ldots, \eta^{\ell-1}$ for E over R is readily seen to be

$$T = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -f_0 & -f_1 & -f_2 & \cdots & -f_{\ell-1} \end{pmatrix},$$

where for $i = 1, \ldots, \ell - 1$, the ith row of T contains a 1 in position $i + 1$, and is zero everywhere else. This matrix is called the **companion matrix of** f. $\square$

Exercise 15.1. Let F be a finite field, and let A be a non-zero $m \times n$ matrix over F. Suppose one chooses a vector $v \in F^{1\times m}$ at random. Show that the probability that vA is the zero vector is at most $1/|F|$.

Exercise 15.2. Design and analyze a probabilistic algorithm that takes as input matrices $A, B, C \in \mathbb{Z}_p^{m\times m}$, where p is a prime. The algorithm should run in time $O(m^2 \operatorname{len}(p)^2)$ and should output either "yes" or "no" so that the following holds:

- if $C = AB$, then the algorithm should always output "yes";
- if $C \neq AB$, then the algorithm should output "no" with probability at least 0.999.

15.3 The inverse of a matrix

Let R be a ring.

For a square matrix $A \in R^{n\times n}$, we call a matrix $X \in R^{n\times n}$ an **inverse** of A if $XA = AX = I$, where I is the $n \times n$ identity matrix.

It is easy to see that if A has an inverse, then the inverse is unique: if X and Y were inverses, then multiplying the equation $I = AY$ on the left by X, we obtain $X = X(AY) = (XA)Y = IY = Y$.

Because the inverse of A is uniquely determined, we denote it by A^{-1}. If A has an inverse, we say that A is **invertible**, or **non-singular**. If A is not invertible, it is sometimes called **singular**. We will use the terms "invertible" and "not invertible." Observe that A is the inverse of A^{-1}; that is, $(A^{-1})^{-1} = A$.

If A and B are invertible $n \times n$ matrices, then so is their product: in fact, it is easy to see that $(AB)^{-1} = B^{-1}A^{-1}$ (verify). It follows that if A is an invertible matrix, and k is a non-negative integer, then A^k is invertible with inverse $(A^{-1})^k$, which we also denote by A^{-k}.

It is also easy to see that A is invertible if and only if the transposed matrix $A^\top$ is invertible, in which case $(A^\top)^{-1} = (A^{-1})^\top$. Indeed, $AX = I = XA$ holds if and only if $X^\top A^\top = I = A^\top X^\top$

The following theorem connects invertibility to linear maps.

Theorem 15.3. *Let $A \in R^{n\times n}$, and let $\rho : R^{1\times n} \to R^{1\times n}$ be the R-linear map that sends $v \in R^{1\times n}$ to vA. Then A is invertible if and only if ρ is bijective.*

Proof. Suppose A is invertible, and let $X \in R^{n\times n}$ be its inverse. The map ρ is surjective, since for any $w \in R^{1\times n}$, $w = wI = wXA = \rho(wX)$. The map ρ is injective, since if $\rho(v) = 0^{1\times n}$, then $v = vI = vAX = \rho(v)X = 0^{1\times n}$.

Suppose ρ is bijective, so that it is an R-module isomorphism. The inverse map ρ^{-1} is also an R-module isomorphism. Let X be the matrix representing ρ^{-1} with respect to the standard basis for $R^{1\times n}$, so that for $w \in R^{1\times n}$, we have $wX = \rho^{-1}(w)$. Since $\rho \circ \rho^{-1} = \rho^{-1} \circ \rho =$ the identity map, it follows that $XA = AX = I$. □

We also have:

Theorem 15.4. *Let $A \in R^{n\times n}$. The following are equivalent:*

(i) A is invertible;

(ii) the rows of A form a basis for $R^{1\times n}$;

(iii) the columns of A form a basis for $R^{n\times 1}$.

Proof. The equivalence of (i) and (ii) follows from the previous theorem, and the fact that the map ρ in that theorem is bijective if and only if the rows of A form a basis for $R^{1\times n}$. The equivalence of (i) and (iii) follows by considering the transpose of A. □

Exercise 15.3. Let R be a ring, and let A be a square matrix over R. Let us call X a **left inverse** of A if $XA = I$, and let us call Y a **right inverse** of A if $AY = I$.

(a) Show that if A has both a left inverse X and a right inverse Y, then $X = Y$ and hence A is invertible.

(b) Assume that R is a field. Show that if A has either a left inverse or a right inverse, then A is invertible.

Note that part (b) of the previous exercise holds for arbitrary rings, but the proof of this is non-trivial, and requires the development of the theory of determinants, which we do not cover in this text.

Exercise 15.4. Show that if A and B are two square matrices over a field such that their product AB is invertible, then both A and B themselves must be invertible.

Exercise 15.5. Show that if A is a square matrix over an arbitrary ring, and A^k is invertible for some $k > 0$, then A is invertible.

15.4 Gaussian elimination

Throughout this section, F denotes a field.

A matrix $B \in F^{m\times n}$ is said to be in **reduced row echelon form** if there exists a sequence of integers $(p_1, \ldots, p_r)$, with $0 \le r \le m$ and $1 \le p_1 < p_2 < \cdots < p_r \le n$, such that the following holds:

- for $i = 1, \ldots, r$, all of the entries in row i of B to the left of entry (i, p_i) are zero (i.e., $B(i, j) = 0$ for $j = 1, \ldots, p_i - 1$);
- for $i = 1, \ldots, r$, all of the entries in column p_i of B above entry (i, p_i) are zero (i.e., $B(i', p_i) = 0$ for $i' = 1, \ldots, i - 1$);
- for $i = 1, \ldots, r$, we have $B(i, p_i) = 1$;
- all entries in rows $r + 1, \ldots, m$ of B are zero (i.e., $B(i, j) = 0$ for $i = r + 1, \ldots, m$ and $j = 1, \ldots, n$).

It is easy to see that if B is in reduced row echelon form, the sequence $(p_1, \ldots, p_r)$ above is uniquely determined, and we call it the **pivot sequence** of B. Several further remarks are in order:

- All of the entries of B are completely determined by the pivot sequence, except for the entries (i, j) with $1 \leq i \leq r$ and $j > i$ with $j \notin \{p_{i+1}, \ldots, p_r\}$, which may be arbitrary.
- If B is an $n \times n$ matrix in reduced row echelon form whose pivot sequence is of length n, then B must be the $n \times n$ identity matrix.
- We allow for an empty pivot sequence (i.e., $r = 0$), which will be the case precisely when $B = 0^{m \times n}$.

***Example* 15.2.** The following 4×6 matrix B over the rational numbers is in reduced row echelon form:

$$B = \begin{pmatrix} 0 & 1 & -2 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The pivot sequence of B is $(2, 4, 5)$. Notice that the first three rows of B are linearly independent, that columns 2, 4, and 5 are linearly independent, and that all of other columns of B are linear combinations of columns 2, 4, and 5. Indeed, if we truncate the pivot columns to their first three rows, we get the 3×3 identity matrix. □

Generalizing the previous example, if a matrix is in reduced row echelon form, it is easy to deduce the following properties, which turn out to be quite useful:

Theorem 15.5. *If B is a matrix in reduced row echelon form with pivot sequence $(p_1, \ldots, p_r)$, then*

(i) rows $1, 2, \ldots, r$ of B are linearly independent;

(ii) columns $p_1, \ldots, p_r$ of B are linearly independent, and all other columns of B can be expressed as linear combinations of columns $p_1, \ldots, p_r$.

Proof. Exercise—just look at the matrix! □

Gaussian elimination is an algorithm that transforms an arbitrary $m \times n$ matrix A into a $m \times n$ matrix B, where B is a matrix in reduced row echelon form obtained from A by a sequence of **elementary row operations**. There are three types of elementary row operations:

Type I: swap two rows,

Type II: multiply a row by a non-zero scalar,

Type III: add a scalar multiple of one row to a different row.

The application of any specific elementary row operation to an $m \times n$ matrix C can be affected by multiplying C on the left by a suitable $m \times m$ matrix M. Indeed, the matrix M corresponding to a particular elementary row operation is simply the matrix obtained by applying the same elementary row operation to the $m \times m$ identity matrix. It is easy to see that for any elementary row operation, the corresponding matrix M is invertible.

We now describe the basic version of Gaussian elimination. The input is an $m \times n$ matrix A.

```
 1.  B ← A, r ← 0
 2.  for j ← 1 to n do
 3.      ℓ ← 0, i ← r
 4.      while ℓ = 0 and i ≤ m do
 5.          i ← i + 1
 6.          if B(i, j) ≠ 0 then ℓ ← i
 7.      if ℓ ≠ 0 then
 8.          r ← r + 1
 9.          swap rows B(r) and B(ℓ)
10.          B(r) ← B(r, j)^{-1} B(r)
11.          for i ← 1 to m do
12.              if i ≠ r then
13.                  B(i) ← B(i) − B(i, j)B(r)
14.  output B
```

The algorithm works as follows. First, it makes a copy B of A (this is not necessary if the original matrix A is not needed afterwards). The algorithm proceeds column by column, starting with the left-most column, so that after processing column j, the first j columns of B are in reduced row echelon form, and the current value of r represents the length of the pivot sequence. To process column j, in steps 3–6 the algorithm first searches for a non-zero element among $B(r+1, j), \ldots, B(m, j)$; if none is found, then the first $j+1$ columns of B are already in reduced row echelon form. Otherwise, one of these non-zero elements is selected as the **pivot element** (the choice is arbitrary), which is then used in steps 8–13 to bring column j into the required form. After incrementing r, the pivot element is brought into position (r, j), using a Type I operation in step 9. Then the entry (r, j) is set to 1, using a Type II operation in step 10. Finally, all the entries above and below entry (r, j) are set to 0, using Type III operations in steps 11–13. Note that because columns $1, \ldots, j-1$ of B were already in reduced row echelon form, none of these operations changes any values in these columns.

As for the complexity of the algorithm, it is easy to see that it performs

$O(mn)$ elementary row operations, each of which takes $O(n)$ operations in F, so a total of $O(mn^2)$ operations in F.

***Example* 15.3.** Consider the execution of the Gaussian elimination algorithm on input

$$A = \begin{pmatrix} [0] & [1] & [1] \\ [2] & [1] & [2] \\ [2] & [2] & [0] \end{pmatrix} \in \mathbb{Z}_3^{3\times 3}.$$

After copying A into B, the algorithm transforms B as follows:

$$\begin{pmatrix} [0] & [1] & [1] \\ [2] & [1] & [2] \\ [2] & [2] & [0] \end{pmatrix} \xrightarrow{B(1)\leftrightarrow B(2)} \begin{pmatrix} [2] & [1] & [2] \\ [0] & [1] & [1] \\ [2] & [2] & [0] \end{pmatrix} \xrightarrow{B(1)\leftarrow [2]B(1)} \begin{pmatrix} [1] & [2] & [1] \\ [0] & [1] & [1] \\ [2] & [2] & [0] \end{pmatrix}$$

$$\xrightarrow{B(3)\leftarrow B(3)-[2]B(1)} \begin{pmatrix} [1] & [2] & [1] \\ [0] & [1] & [1] \\ [0] & [1] & [1] \end{pmatrix} \xrightarrow{B(1)\leftarrow B(1)-[2]B(2)} \begin{pmatrix} [1] & [0] & [2] \\ [0] & [1] & [1] \\ [0] & [1] & [1] \end{pmatrix}$$

$$\xrightarrow{B(3)\leftarrow B(3)-B(2)} \begin{pmatrix} [1] & [0] & [2] \\ [0] & [1] & [1] \\ [0] & [0] & [0] \end{pmatrix}$$

□

Suppose the Gaussian elimination algorithm performs a total of t elementary row operations. Then as discussed above, the application of the eth elementary row operation, for $e = 1, \ldots, t$, amounts to multiplying the current value of the matrix B on the left by a particular invertible $m \times m$ matrix M_e. Therefore, the final, output value of B satisfies the equation

$$B = MA \quad \text{where} \quad M = M_t M_{t-1} \cdots M_1.$$

Since the product of invertible matrices is also invertible, we see that M itself is invertible.

Although the algorithm as presented does not compute the matrix M, it can be easily modified to do so. The resulting algorithm, which we call **extended Gaussian elimination**, is the same as plain Gaussian elimination, except that we initialize the matrix M to be the $m \times m$ identity matrix, and we add the following steps:

- Just before step 9, we add the step: swap rows $M(r)$ and $M(\ell)$.
- Just before step 10, we add the step: $M(r) \leftarrow B(r, j)^{-1} M(r)$.
- Just before step 13, we add the step: $M(i) \leftarrow M(i) - B(i, j) M(r)$.

At the end of the algorithm we output M in addition to B.

So we simply perform the same elementary row operations on M that we perform on B. The reader may verify that the above algorithm is correct, and that it uses $O(mn(m+n))$ operations in F.

***Example* 15.4.** Continuing with Example 15.3, the execution of the extended Gaussian elimination algorithm initializes M to the identity matrix, and then transforms M as follows:

$$\begin{pmatrix} [1] & [0] & [0] \\ [0] & [1] & [0] \\ [0] & [0] & [1] \end{pmatrix} \xrightarrow{M(1) \leftrightarrow M(2)} \begin{pmatrix} [0] & [1] & [0] \\ [1] & [0] & [0] \\ [0] & [0] & [1] \end{pmatrix} \xrightarrow{M(1) \leftarrow [2]M(1)} \begin{pmatrix} [0] & [2] & [0] \\ [1] & [0] & [0] \\ [0] & [0] & [1] \end{pmatrix}$$

$$\xrightarrow{M(3) \leftarrow M(3) - [2]M(1)} \begin{pmatrix} [0] & [2] & [0] \\ [1] & [0] & [0] \\ [0] & [2] & [1] \end{pmatrix} \xrightarrow{M(1) \leftarrow M(1) - [2]M(2)} \begin{pmatrix} [1] & [2] & [0] \\ [1] & [0] & [0] \\ [0] & [2] & [1] \end{pmatrix}$$

$$\xrightarrow{M(3) \leftarrow M(3) - M(2)} \begin{pmatrix} [1] & [2] & [0] \\ [1] & [0] & [0] \\ [2] & [2] & [1] \end{pmatrix}$$

□

Exercise 15.6. For each type of elementary row operation, describe the matrix M which corresponds to it, as well as M^{-1}.

Exercise 15.7. Given a matrix $B \in F^{m \times n}$ in reduced row echelon form, show how to compute its pivot sequence using $O(n)$ operations in F.

Exercise 15.8. In §4.4, we saw how to speed up matrix multiplication over $\mathbb{Z}$ using the Chinese remainder theorem. In this exercise, you are to do the same, but for performing Gaussian elimination over $\mathbb{Z}_p$, where p is a large prime. Suppose you are given an $m \times m$ matrix A over $\mathbb{Z}_p$, where $\mathrm{len}(p) = \Theta(m)$. Straightforward application of Gaussian elimination would require $O(m^3)$ operations in $\mathbb{Z}_p$, each of which takes time $O(m^2)$, leading to a total running time of $O(m^5)$. Show how to use the techniques of §4.4 to reduce the running time of Gaussian elimination to $O(m^4)$.

15.5 Applications of Gaussian elimination

Throughout this section, A is an arbitrary $m \times n$ matrix over a field F, and $MA = B$, where M is an invertible $m \times m$ matrix, and B is in reduced row echelon form with pivot sequence $(p_1, \ldots, p_r)$. This is precisely the information produced by the extended Gaussian elimination algorithm, given

A as input (the pivot sequence can easily be "read" directly from B—see Exercise 15.7).

Let $V := F^{1\times m}$, $W := F^{1\times n}$, and $\rho : V \to W$ be the F-linear map that sends $v \in V$ to $vA \in W$.

Computing the image and kernel

Consider first the **row space** of A, that is, the vector space spanned by the rows of A, or equivalently, the image of ρ.

We claim that the row space of A is the same as the row space of B. To see this, note that for any $v \in V$, since $B = MA$, we have $vB = v(MA) = (vM)A$, and so the row space of B is contained in the row space of A. For the other containment, note that since M is invertible, we can write $A = M^{-1}B$, and apply the same argument.

Further, note that row space of B, and hence that of A, clearly has dimension r. Indeed, as stated in Theorem 15.5, the first r rows of B form a basis for the row space of B.

Consider next the kernel of ρ, or what we might call the **row null space** of A. We claim that the last $m - r$ rows of M form a basis for $\ker(\rho)$. Clearly, just from the fact that $MA = B$ and the fact that the last $m - r$ rows of B are zero, it follows that the last $m - r$ rows of M are contained in $\ker(\rho)$. Furthermore, as M is invertible, its rows form a basis for V (see Theorem 15.4), and so in particular, they are linearly independent. It therefore suffices to show that the last $m - r$ rows of M span the entire kernel. Now, suppose there were a vector $v \in \ker(\rho)$ outside the subspace spanned by the last $m - r$ rows of M. As the rows of M span V, we may write $v = a_1M(1) + \cdots + a_mM(m)$, where $a_i \neq 0$ for some $i = 1, \ldots, r$. Setting $\tilde{v} := (a_1, \ldots, a_m)$, we see that $v = \tilde{v}M$, and so

$$\rho(v) = vA = (\tilde{v}M)A = \tilde{v}(MA) = \tilde{v}B,$$

and from the fact that the first r rows of B are linearly independent and the last $m - r$ rows of B are zero, we see that $\tilde{v}B$ is not the zero vector (and because $\tilde{v}$ has a non-zero entry in one its first r positions). We have derived a contradiction, and hence may conclude that the last $m - r$ rows of M span $\ker(\rho)$.

Finally, note that if $m = n$, then A is invertible if and only if its row space has dimension m, which holds if and only if $r = m$, and in the latter case, B will be the identity matrix, and hence M is the inverse of A.

Let us summarize the above discussion:

- *The first r rows of B form a basis for the row space of A (i.e., the image of ρ).*
- *The last $m - r$ rows of M form a basis for the row null space of A (i.e., the kernel of ρ).*
- *If $m = n$, then A is invertible (i.e., ρ is an isomorphism) if and only if $r = m$, in which case M is the inverse of A (i.e., the matrix representing ρ^{-1}).*

So we see that from the output of the extended Gaussian elimination algorithm, we can simply "read off" bases for both the image and the kernel, as well as the inverse (if it exists), of a linear map represented as a matrix with respect to some ordered bases. Also note that this procedure provides a "constructive" version of Theorem 14.29.

***Example* 15.5.** Continuing with Examples 15.3 and 15.4, we see that the vectors $([1], [0], [2])$ and $([0], [1], [1])$ form a basis for the row space of A, while the vector $([2], [2], [1])$ is a basis for the row null space of A. □

Solving linear systems of equations

Suppose that in addition to the matrix A, we are given $w \in W$, and want to find a solution v (or perhaps describe all solutions v), to the equation

$$vA = w. \tag{15.1}$$

Equivalently, we can phrase the problem as finding an element (or describing all elements) of the set $\rho^{-1}(w)$.

Now, if there exists a solution at all, say $v \in V$, then since $\rho(v) = \rho(\tilde{v})$ if and only if $v \equiv \tilde{v} \pmod{\ker(\rho)}$, it follows that the set of all solutions to (15.1) is equal to the coset $v + \ker(\rho)$. Thus, given a basis for $\ker(\rho)$ and any solution v to (15.1), we have a complete and concise description of the set of solutions to (15.1).

As we have discussed above, the last $m - r$ rows of M give us a basis for $\ker(\rho)$, so it suffices to determine if $w \in \mathrm{img}(\rho)$, and if so, determine a single pre-image v of w.

Also as we discussed, $\mathrm{img}(\rho)$, that is, the row space of A, is equal to the row space of B, and because of the special form of B, we can quickly and easily determine if the given w is in the row space of B, as follows. By definition, w is in the row space of B iff there exists a vector $\bar{v} \in V$ such that $\bar{v}B = w$. We may as well assume that all but the first r entries of $\bar{v}$ are zero. Moreover, $\bar{v}B = w$ implies that for $i = 1, \ldots, r$, the ith entry if $\bar{v}$ is equal to the p_ith entry of w. Thus, the vector $\bar{v}$, if it exists, is completely

determined by the entries of w at positions $p_1, \ldots, p_r$. We can construct $\bar{v}$ satisfying these conditions, and then test if $\bar{v}B = w$. If not, then we may conclude that (15.1) has no solutions; otherwise, setting $v := \bar{v}M$, we see that $vA = (\bar{v}M)A = \bar{v}(MA) = \bar{v}B = w$, and so v is a solution to (15.1).

One easily verifies that if we implement the above procedure as an algorithm, the work done in addition to running the extended Gaussian elimination algorithm amounts to $O(m(n+m))$ operations in F.

A special case of the above procedure is when $m = n$ and A is invertible, in which case (15.1) has a unique solution, namely, $v := wM$, since in this case, $M = A^{-1}$.

The rank of a matrix

Define the **row rank** of A to be the dimension of its row space, which is $\dim_F(\mathrm{img}(\rho))$, and define the **column rank** of A to be the dimension of its **column space**, that is, the space spanned by the columns of A.

Now, the column space A may not be the same as the column space of B, but from the relation $B = MA$, and the fact that M is invertible, it easily follows that these two subspaces are isomorphic (via the isomorphism that sends v to Mv), and hence have the same dimension. Moreover, by Theorem 15.5, the column rank of B is r, which is the same as the row rank of A.

So we may conclude: *The column rank and row rank of A are the same.*

Because of this, we define the **rank** of a matrix to be the common value of its row and column rank.

The orthogonal complement of a subspace

So as to give equal treatment to rows and columns, one can also define the **column null space** of A to be the kernel of the linear map defined by multiplication on the left by A. By applying the results above to the transpose of A, we see that the column null space of A has dimension $n - r$, where r is the rank of A.

Let $U \subseteq W$ denote the row space of A, and let $\bar{U} \subseteq W$ denote the set of all vectors $\bar{u} \in W$ whose transpose $\bar{u}^\top$ belong to the column null space of A. Now, U is a subspace of W of dimension r and $\bar{U}$ is a subspace of W of dimension $n - r$.

Moreover, if $U \cap \bar{U} = \{0_V\}$, then by Theorem 14.13 we have an isomorphism of $U \times \bar{U}$ with $U + \bar{U}$, and since $U \times \bar{U}$ has dimension n, it must be the

case that $U + \bar{U} = W$. It follows that every element of W can be expressed uniquely as $u + \bar{u}$, where $u \in U$ and $\bar{u} \in \bar{U}$.

Now, all of the conclusions in the previous paragraph hinged on the assumption that $U \cap \bar{U} = \{0_V\}$. The space $\bar{U}$ consists precisely of all vectors $\bar{u} \in W$ which are "orthogonal" to all vectors $u \in U$, in the sense that the "inner product" $u\bar{u}^\top$ is zero. For this reason, $\bar{U}$ is sometimes called the "orthogonal complement of U." The condition $U \cap \bar{U} = \{0_V\}$ is equivalent to saying that U contains no non-zero "self-orthogonal vectors" u such that $uu^\top = 0_F$. If F is the field of real numbers, then of course there are no non-zero self-orthogonal vectors, since $uu^\top$ is the sum of the squares of the entries of u. However, for other fields, there may very well be non-zero self-orthogonal vectors. As an example, if $F = \mathbb{Z}_2$, then any vector u with an even number of 1-entries is self orthogonal.

So we see that while much of the theory of vector spaces and matrices carries over without change from familiar ground fields, like the real numbers, to arbitrary ground fields F, not everything does. In particular, the usual decomposition of a vector space into a subspace and its orthogonal complement breaks down, as does any other procedure that relies on properties specific to "inner product spaces."

For the following three exercises, as above, A is an arbitrary $m \times n$ matrix over a field F, and $MA = B$, where M is an invertible $m \times m$ matrix, and B is in reduced row echelon form.

Exercise 15.9. Show that the column null space of A is the same as the column null space of B.

Exercise 15.10. Show how to compute a basis for the column null space of A using $O(r(n-r))$ operations in F, given A and B.

Exercise 15.11. Show that the matrix B is uniquely determined by A; more precisely, show that if $M'A = B'$, where M' is an invertible $m \times m$ matrix, and B' is in reduced row echelon form, then $B' = B$.

In the following two exercises, the theory of determinants could be used; however, they can all be solved directly, without too much difficulty, using just the ideas developed so far in the text.

Exercise 15.12. Let p be a prime. A matrix $A \in \mathbb{Z}^{m \times m}$ is called *invertible modulo p* if and only if there exists a matrix $X \in \mathbb{Z}^{m \times m}$ such that $AX \equiv XA \equiv I \pmod{p}$, where I is the $m \times m$ integer identity matrix. Here, two matrices are considered congruent with respect to a given modulus if and

only if their corresponding entries are congruent. Show that A is invertible modulo p if and only if

- A is invertible over $\mathbb{Q}$, and
- the entries of A^{-1} lie in $\mathbb{Q}^{(p)}$ (see Example 9.23).

Exercise 15.13. You are given a matrix $A \in \mathbb{Z}^{m \times m}$ and a prime p such that A is invertible modulo p. Suppose that you are also given $w \in \mathbb{Z}^{1 \times m}$.

(a) Show how to efficiently compute a vector $v \in \mathbb{Z}^{1 \times m}$ such that $vA = w \pmod{p}$, and that v is uniquely determined modulo p.

(b) Given a vector v as in part (a), along with an integer $e \geq 1$, show how to efficiently compute $\hat{v} \in \mathbb{Z}^{1 \times m}$ such that $\hat{v}A = w \pmod{p^e}$, and that $\hat{v}$ is uniquely determined modulo p^e. Hint: mimic the "lifting" procedure discussed in §13.3.2.

(c) Using parts (a) and (b), design and analyze an efficient algorithm that takes the matrix A and the prime p as input, together with a bound H on the absolute value of the numerator and denominator of the entries of the vector v' that is the unique (rational) solution to the equation $v'A = w$. Your algorithm should run in time polynomial in the length of H, the length of p, and the sum of the lengths of the entries of A and w. Hint: use rational reconstruction, but be sure to fully justify its application.

Note that in the previous exercise, one can use the theory of determinants to derive good bounds, in terms of the lengths of the entries of A and w, on the size of the least prime p such that A is invertible modulo p (assuming A is invertible over the rationals), and the length of the numerator and denominator of the entries of rational solution v' to the equation $v'A = w$. The interested reader who is familiar with the basic theory of determinants is encouraged to establish such bounds.

The next two exercises illustrate how Gaussian elimination can be adapted, in certain cases, to work in rings that are not necessarily fields. Let R be an arbitrary ring. A matrix $B \in R^{m \times n}$ is said to be in **row echelon form** if there exists a pivot sequence $(p_1, \ldots, p_r)$, with $0 \leq r \leq m$ and $1 \leq p_1 < p_2 < \cdots < p_r \leq n$, such that the following holds:

- for $i = 1, \ldots, r$, all of the entries in row i of B to the left of entry (i, p_i) are zero;
- for $i = 1, \ldots, r$, we have $B(i, p_i) \neq 0$;
- all entries in rows $r + 1, \ldots, m$ of B are zero.

Exercise 15.14. Let R be the ring $\mathbb{Z}_{p^e}$, where p is prime and $e > 1$. Let $\pi := [p] \in R$. The goal of this exercise is to develop an efficient algorithm for the following problem: given a matrix $A \in R^{m \times n}$, with $m > n$, find a vector $v \in R^{1 \times m}$ such that $vA = 0^{1 \times n}$ but $v \notin \pi R^{1 \times m}$.

(a) Show how to modify the extended Gaussian elimination algorithm to solve the following problem: given a matrix $A \in R^{m \times n}$, compute $M \in R^{m \times m}$ and $B \in R^{m \times n}$, such that $MA = B$, M is invertible, and B is in row echelon form. Your algorithm should run in time $O(mn(m+n)e^2 \operatorname{len}(p)^2)$. Assume that the input includes the values p and e. Hint: when choosing a pivot element, select one divisible by a minimal power of π; as in ordinary Gaussian elimination, your algorithm should only use elementary row operations to transform the input matrix.

(b) Using the fact that the matrix M computed in part (a) is invertible, argue that none of its rows belong to $\pi R^{1 \times m}$.

(c) Argue that if $m > n$ and the matrix B computed in part (a) has pivot sequence $(p_1, \ldots, p_r)$, then $m - r > 0$ and if v is any one of the last $m - r$ rows of M, then $vA = 0^{1 \times n}$.

(d) Give an example that shows that the first r rows of B need not be linearly independent and that the last $m - r$ rows of M need not span the kernel of the R-linear map that sends $w \in R^{1 \times m}$ to $wA \in R^{1 \times n}$.

Exercise 15.15. Let R be the ring $\mathbb{Z}_\ell$, where $\ell > 1$ is an integer. You are given a matrix $A \in R^{m \times n}$. Show how to efficiently compute $M \in R^{m \times m}$ and $B \in R^{m \times n}$ such that $MA = B$, M is invertible, and B is in row echelon form. Your algorithm should run in time $O(mn(m+n)\operatorname{len}(\ell)^2)$. Hint: to zero-out entries, you should use "rotations"—for integers a, b, d, s, t with

$$d = \gcd(a, b) \neq 0 \quad \text{and} \quad as + bt = d,$$

and for row indices r, i, a rotation simultaneously updates rows r and i of a matrix C as follows:

$$(C(r), C(i)) \leftarrow (sC(r) + tC(i), -\frac{b}{d}C(r) + \frac{a}{d}C(i));$$

observe that if $C(r, j) = [a]_\ell$ and $C(i, j) = [b]_\ell$ before applying the rotation, then $C(r, j) = [d]_\ell$ and $C(i, j) = [0]_\ell$ after the rotation.

15.6 Notes

While a trivial application of the defining formulas yields a simple algorithm for multiplying two $m \times m$ matrices over a ring R that uses $O(m^3)$ operations

in R, this algorithm is not the best, asymptotically speaking. The currently fastest algorithm for this problem, due to Coppersmith and Winograd [28], uses $O(m^\omega)$ operations in R, where $\omega < 2.376$. We note, however, that the good old $O(m^3)$ algorithm is still the only one used in almost any practical setting.