1

Preliminaries

This chapter is dedicated to establishing the notation and terminology that we will use throughout this book. More specific concepts and definitions will be introduced as and when needed. None of the material in this chapter is intended to be of an expository nature. Rather, it collects together basic notions and facts from linear algebra, real analysis, and fundamentals of computation that are required for our study in later chapters. No proofs of these facts are provided here, and instead we refer the reader to the references listed at the end of this chapter. The handful of exercises in this chapter are included primarily because they will be relevant later. They are in no way meant to be a comprehensive test of familiarity with these subjects.

We will use $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ to denote the set of natural numbers (starting at 1), the set of integers, the set of rational numbers, and the set of real numbers, respectively. $\mathbb{Z}_+, \mathbb{Q}_+, \mathbb{R}_+$ will denote the nonnegative integers, rationals, and reals, respectively. For any real number $r \in \mathbb{R}$, $|r|$ will denote the absolute value of r. We denote the greatest integer smaller than or equal to r by $\lfloor r \rfloor$, the smallest integer greater than or equal to r by $\lceil r \rceil$, $\{r\}$ will be used to denote $r - \lfloor r \rfloor$, and $\lfloor r \rceil$ will denote the integer closest to r in absolute value. $\ln(r)$ will denote the natural logarithm of r, and $\log(r)$ will denote the logarithm of r in base 2. Logarithms in any other base b will be explicitly denoted by $\log_b(r)$.

δ_{ij} will denote the standard *Kronecker delta function*, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise (the range of the indices i, j will be clear from context). For any set X, $\#(X)$ will denote the cardinality, i.e., number of elements in X (possibly $+\infty$).[1] 2^X will denote the power set of X. We will also use the following standard asymptotic notations of $O(\cdot)$ and $\Omega(\cdot)$: Given two functions $F(p_1, \ldots, p_k), G(p_1, \ldots, p_k)$ of nonnegative, real-valued

[1] We will not worry about different cardinal numbers beyond countable and uncountable sets. When referring to cardinality, $+\infty$ will be used for both countable and uncountable sets.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

parameters $p_1, \ldots, p_k$, we say $F = O(G)$ (informally, "F is asymptotically upper bounded by G") and $G = \Omega(F)$ (informally, "G is asymptotically lower bounded by F") if there exist constants $C > 0, M > 0$ such that $|F(p_1, \ldots, p_k)| \leq C|G(p_1, \ldots, p_k)|$ for all $p_1, \ldots, p_k \in \mathbb{R}_+$ with $p_i > M$ for all $i \in \{1, \ldots, k\}$.[2] We mention here that while this version of asymptotic notation for functions with multiple arguments will suffice for our purposes, there are other ways to tackle asymptotics with multiple parameters; see [79] and [144] for a discussion of the subtleties that can arise.

1.1 Euclidean Spaces

All of the action in this book will be in finite-dimensional Euclidean spaces. For any $d \in \mathbb{N}$, we use $\mathbb{R}^d$ to denote the d-dimensional Euclidean space, i.e., $\mathbb{R}^d$ is the set of all d-tuples $\mathbf{x} = (x_1, \ldots, x_d)$ of real numbers; these are also called *vectors* or *points* in $\mathbb{R}^d$. Vectors will be boldface and scalars will be nonbold italics. Subscripts will be used to denote the coordinates, i.e., $\mathbf{x}_i := x_i$. We will use the notation $\mathbb{R}^d_+$ to denote the set of all vectors with nonnegative coordinates. The symbol $\mathbf{e}^i$, $i = 1, \ldots, d$, will denote the ith unit vector, i.e., the vector which has 1 in the ith coordinate and 0 in every other coordinate; to emphasize the dimension we will sometimes use $\mathbf{e}^i_d$. $\mathbf{0}$ and $\mathbf{1}$ will denote the vector of all zeros and all ones, respectively; to emphasize the dimension we will sometimes use $\mathbf{0}_d$ and $\mathbf{1}_d$. $\mathbf{0}$ is also called the *origin*. We will use the coordinate-wise *addition* operation

$$\mathbf{x} + \mathbf{y} := (\mathbf{x}_1 + \mathbf{y}_1, \ldots, \mathbf{x}_d + \mathbf{y}_d) \quad \text{for any } \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,$$

and the operation of *scalar multiplication*

$$\alpha\mathbf{x} := (\alpha\mathbf{x}_1, \ldots, \alpha\mathbf{x}_d) \quad \text{for any } \alpha \in \mathbb{R} \text{ and } \mathbf{x} \in \mathbb{R}^d.$$

These operations impose a vector space structure on $\mathbb{R}^d$. Correspondingly, a subset $L \subseteq \mathbb{R}^d$ is called a *linear subspace* if L is closed under the above operations of addition and scalar multiplication. It is a convention to not consider the empty set as a subspace. Thus, the smallest linear subspace is $\{\mathbf{0}\}$.

Definition 1.1.1 A *norm on* $\mathbb{R}^d$ is a function $N : \mathbb{R}^d \to \mathbb{R}_+$ satisfying the following properties:

[2] Sometimes, it is more convenient to present asymptotic arguments when some parameters approach 0 as a limit, instead of $+\infty$ as in the definition above. This should be clear from context.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

1. $N(\mathbf{x}) = 0$ if and only if $\mathbf{x} = \mathbf{0}$,
2. $N(\alpha\mathbf{x}) = |\alpha|N(\mathbf{x})$ for all $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^d$,
3. $N(\mathbf{x}+\mathbf{y}) \leq N(\mathbf{x}) + N(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. (Triangle inequality)

Example 1.1.2 For any real number $p \geq 1$, define the ℓ^p norm on $\mathbb{R}^d$ as $\|\mathbf{x}\|_p = (|\mathbf{x}_1|^p + |\mathbf{x}_2|^p + \cdots + |\mathbf{x}_d|^p)^{\frac{1}{p}}$. $p = 2$ is also called the *standard Euclidean norm*; we will drop the subscript 2 to denote the standard norm: $\|\mathbf{x}\| = \sqrt{\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2}$. The ℓ^∞ norm is defined as $\|\mathbf{x}\|_\infty = \max_{i=1}^d |\mathbf{x}_i|$.

The following is an important result that shows that in many situations, the specific choice of a norm does not matter.

Theorem 1.1.3 (Equivalence of norms) *Let N, N' be any two norms on $\mathbb{R}^d$. There exists a constant $C > 0$ (depending on N and N') such that $N'(\mathbf{x}) \leq C \cdot N(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$.*

Definition 1.1.4 Any norm on $\mathbb{R}^d$ defines a distance between points in $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ as $d_N(\mathbf{x}, \mathbf{y}) := N(\mathbf{x} - \mathbf{y})$. This is called the *metric* or *distance induced by the norm*. Such a metric satisfies three important properties:

1. $d_N(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$,
2. $d_N(\mathbf{x}, \mathbf{y}) = d_N(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$,
3. $d_N(\mathbf{x}, \mathbf{z}) \leq d_N(\mathbf{x}, \mathbf{y}) + d_N(\mathbf{y}, \mathbf{z})$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$. (Triangle inequality)

We will often identify the set $\mathbb{R}^n \times \mathbb{R}^d$ with the Euclidean space $\mathbb{R}^{n+d}$ by "concatenation," i.e., given $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^d$, we will associate the vector $(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}_d) \in \mathbb{R}^{n+d}$ with the tuple $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^d$. Given two norms N on $\mathbb{R}^n$ and N' on $\mathbb{R}^d$, $(\mathbf{x}, \mathbf{y}) \mapsto N(\mathbf{x}) + N'(\mathbf{y})$ can be verified to be a new norm on $\mathbb{R}^d \times \mathbb{R}^d \equiv \mathbb{R}^{n+d}$.

Definition 1.1.5 We also utilize the (standard) inner product of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$: $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}_1\mathbf{y}_1 + \mathbf{x}_2\mathbf{y}_2 + \cdots + \mathbf{x}_d\mathbf{y}_d$. (Note that $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle$). We say $\mathbf{x}$ and $\mathbf{y}$ are *orthogonal* if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. A set of vectors $\mathbf{x}^1, \ldots, \mathbf{x}^k$ is said to be *orthonormal* if $\langle \mathbf{x}^i, \mathbf{x}^j \rangle = \delta_{ij}$ for all $i, j \in \{1, \ldots, k\}$.

Theorem 1.1.6 (Cauchy–Schwarz inequality) *For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \cdot \|\mathbf{y}\|$.*

Definition 1.1.7 For any norm N and $\mathbf{x} \in \mathbb{R}^d$, $r \in \mathbb{R}_+$, we will call the set $B_N(\mathbf{x}, r) := \{\mathbf{y} \in \mathbb{R}^d : N(\mathbf{y} - \mathbf{x}) \leq r\}$ as the *ball around $\mathbf{x}$ of radius r*. $B_N(\mathbf{0}, 1)$ will be called the *unit ball for the norm N*. We will drop the subscript N when we speak of the standard Euclidean norm and there is no chance of confusion in the context.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

A subset $X \subseteq \mathbb{R}^d$ is said to be *bounded* if there exists $R \geq 0$ such that $X \subseteq B(\mathbf{0}, R)$.

Definition 1.1.8 (Set operations) Given any set $X \subseteq \mathbb{R}^d$ and a scalar $\alpha \in \mathbb{R}$,

$$\alpha X := \{\alpha \mathbf{x} : \mathbf{x} \in X\}.$$

Given any two sets $X, Y \subseteq \mathbb{R}^d$, we define the *Minkowski sum* of X, Y as

$$X + Y := \{\mathbf{x} + \mathbf{y} : \mathbf{x} \in X, \mathbf{y} \in Y\}.$$

We will write $X + \mathbf{t}$ if $Y = \{\mathbf{t}\}$ is a singleton; this is called the *translate of* X *by* $\mathbf{t}$.

The *characteristic function*[3] *of* X is defined as $\mathbb{1}_X(\mathbf{x}) := \begin{cases} 1 & \text{if } \mathbf{x} \in X, \\ 0 & \text{otherwise.} \end{cases}$

1.2 Linear Algebra

An important tool in linear algebra is the notion of linear independence.

Definition 1.2.1 Let $\mathbf{x}^1, \ldots, \mathbf{x}^k \in \mathbb{R}^d$. A *linear combination of* $\mathbf{x}^1, \ldots, \mathbf{x}^k$ is any point of the form $\lambda_1 \mathbf{x}_1 + \cdots + \lambda_k \mathbf{x}_k$ where $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$. The set of all linear combinations of $\mathbf{x}^1, \ldots, \mathbf{x}^k$ will be denoted by $\operatorname{span}(\mathbf{x}^1, \ldots, \mathbf{x}^k)$.

A collection of points $\mathbf{x}^1, \ldots, \mathbf{x}^n \in \mathbb{R}^d$ is said to be *linearly independent* if there is no $i \in \{1, \ldots, n\}$ such that $\mathbf{x}^i$ can be expressed as a linear combination of the remaining $\mathbf{x}^j$, $j \neq i$. If there exists such an index i, then $\mathbf{x}^1, \ldots, \mathbf{x}^n$ are said to be *linearly dependent*.

Similarly, a set $X \subseteq \mathbb{R}^d$ (possibly infinite) is said to be linearly independent if no $\mathbf{x} \in X$ can be expressed as a linear combination of (finitely) many points in $X \setminus \{\mathbf{x}\}$; otherwise, X is said to be linearly dependent.

According to the above definition, any set containing the origin and another nonzero vector is linearly dependent. We adopt the standard convention that even the singleton set containing the origin is a linearly dependent set. Then we have the following equivalent formulation of linear independence, which is very useful.

Proposition 1.2.2 *Points* $\mathbf{x}^1, \ldots, \mathbf{x}^k \in \mathbb{R}^d$ *are linearly independent if and only if* $\lambda_1 \mathbf{x}_1 + \cdots + \lambda_k \mathbf{x}_k = \mathbf{0}$ *implies* $\lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$.

A fundamental object of study in linear algebra is the notion of a linear transformation.

[3] The alternative terminology "indicator function" will be used for a different but related concept in this book (Example 3.1.11), in line with the practice in convex analysis.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

Definition 1.2.3 A *linear map or transformation or function* is a function $T : \mathbb{R}^d \to \mathbb{R}^m$ between two Euclidean spaces such that $T(\lambda \mathbf{x} + \gamma \mathbf{y}) = \lambda T(\mathbf{x}) + \gamma T(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\lambda, \gamma \in \mathbb{R}$.

The above definition is "coordinate free," and so it can be extended to maps $T : L \to L'$ where $L \subseteq \mathbb{R}^d$ and $L' \subseteq \mathbb{R}^m$ are arbitrary subspaces. This will be useful at a few places in the book (e.g., projections onto linear subspaces).

Any linear transformation between $\mathbb{R}^d$ and $\mathbb{R}^m$ can be represented by an $m \times d$ matrix. In other words, for any linear transformation $T : \mathbb{R}^d \to \mathbb{R}^m$, there exists an $m \times d$ matrix A such that $T(\mathbf{x}) = A\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^d$, where we view $\mathbf{x}$ as a matrix with a single column and $A\mathbf{x}$ denotes standard matrix multiplication. In particular, $A_{ij} := T(\mathbf{e}^j)_i$. Conversely, any $m \times d$ matrix A gives a linear transformation $\mathbf{x} \mapsto A\mathbf{x}$. This also leads to the observation that the matrix corresponding to a composition of linear transformations is the product of the matrices corresponding to the individual transformations. We will also consider "translated" versions of linear transformations.

Definition 1.2.4 An *affine map* or *transformation* or *function* is a function $T : \mathbb{R}^d \to \mathbb{R}^m$ of the form $T(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, where A is an $m \times d$ matrix and $\mathbf{b} \in \mathbb{R}^m$.

Given a set $X \subseteq \mathbb{R}^d$, we will use the notation $T(X)$ to denote the image of this set under the affine map T; similarly $A(X) := \{A\mathbf{x} : \mathbf{x} \in X\}$. The set of all $m \times d$ matrices will be denoted by $\mathbb{R}^{m \times d}$. We use the notation $\mathbf{0}_{m \times d}$ to denote the $m \times d$ matrix with all zero entries. The *rank of a matrix* A will be denoted by $\mathrm{rk}(A)$ – it is the maximum number of linearly independent rows of A, which is equal to the maximum number of linearly independent columns of A. A matrix is said to have *full row rank* (respectively, *full column rank*) if its rank equals the number of rows (respectively, the number of columns). Any affine transformation $A\mathbf{x} + \mathbf{b}$ is also said to have rank $\mathrm{rk}(A)$. The *transpose* A^T of $A \in \mathbb{R}^{m \times d}$ is defined by $A^T_{ij} = A_{ji}$ for all $i \in \{1, \ldots, d\}$ and $j \in \{1, \ldots, m\}$. The following characterization of the transpose will be important for us.

Proposition 1.2.5 *Let $A \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{d \times m}$. Then $\langle \mathbf{y}, A\mathbf{x} \rangle = \langle B\mathbf{y}, \mathbf{x} \rangle$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$ if and only if $B = A^T$.*

The above also shows that $(AB)^T = B^T A^T$ for any two matrices with the appropriate dimensions.

When $m = d$, we say that the matrix is *square*. For a square matrix, A_{ii} are called the *diagonal entries*. A *diagonal matrix* is a square matrix A such that $A_{ij} = 0$ if $i \neq j$; We use I_d to denote the $d \times d$ *identity matrix*, i.e., the matrix where the diagonal elements are 1 and all other entries are 0. The function

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

$$\det(A) := \sum_{\substack{\text{Permutations} \\ \sigma\colon \{1,\dots,d\} \to \{1,\dots,d\}}} \operatorname{sgn}(\sigma) \prod_{i=1}^{d} A_{i\sigma(i)}$$

defined on the space of all $d \times d$ square matrices is called the *determinant of* A, where $\operatorname{sgn}(\sigma)$ is the signature of the permutation σ.[4] The following formula for computing determinants of products of matrices is very useful.

Theorem 1.2.6 (Cauchy–Binet formula) *Let $m, d \geq 1$ and let $A \in \mathbb{R}^{m\times d}$ and $B \in \mathbb{R}^{d\times m}$. Then*

$$\det(AB) = \sum_{\substack{S \subseteq \{1,\dots,d\} \\ \#(S) = m}} \det(A_S)\det(B_S),$$

where A_S is the $m \times m$ submatrix of A with columns indexed by S and B_S is the $m \times m$ submatrix of B with rows indexed by S.

Theorem 1.2.7 *Let $T\colon \mathbb{R}^d \to \mathbb{R}^d$ be a linear transformation, and let $A \in \mathbb{R}^{d\times d}$ be the corresponding matrix. The following are equivalent.*

1. *T is injective, i.e., $T(\mathbf{x}) = T(\mathbf{y})$ implies $\mathbf{x} = \mathbf{y}$.*
2. *T is onto, i.e., $T(\mathbb{R}^d) = \mathbb{R}^d$.*
3. *T is bijective and T^{-1} is a linear map as well.*
4. *There exists a matrix $B \in \mathbb{R}^{d\times d}$ such that $AB = BA = I_d$. B is denoted by A^{-1} and is called* the inverse matrix of A.
5. *$\det(A) \neq 0$.*

A matrix satisfying part 4. of Theorem 1.2.7 is called an *invertible matrix*. Theorem 1.2.7 also shows that A^{-1} is the matrix corresponding to the linear map T^{-1}, and $(AB)^{-1} = B^{-1}A^{-1}$. The following formula relating the inverse and transpose is useful.

Proposition 1.2.8 *Let $A \in \mathbb{R}^{d\times d}$. Then A is invertible if and only if A^T is invertible. Moreover, $(A^T)^{-1} = (A^{-1})^T$. We will use the shorthand A^{-T} to denote this matrix.*

The following theorem singles out a special class of matrices that are important in various contexts.

Theorem 1.2.9 *Let $A \in \mathbb{R}^{d\times d}$ be a square matrix. The following are all equivalent.*

[4] The signature of a permutation is +1 if it has an even number of inversions, and −1 if it has an odd number of inversions.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

1. *The columns of A are orthonormal.*
2. *The rows of A are orthonormal.*
3. $A^{-1} = A^T$.
4. $\langle A\mathbf{x}, A\mathbf{y}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.
5. $\|A\mathbf{x}\| = \|\mathbf{x}\|$ *for all* $\mathbf{x} \in \mathbb{R}^d$.

Matrices satisfying the conditions in Theorem 1.2.9 are called *orthonormal matrices*.

We now come to a fundamental theorem of linear algebra.

Theorem 1.2.10 *Let $X \subseteq \mathbb{R}^d$. The following are equivalent.*

1. *X is a linear subspace.*
2. *There exists $0 \leq m \leq d$ and linearly independent vectors $\mathbf{v}^1, \ldots, \mathbf{v}^m \in X$ such that, $X = \text{span}(\{\mathbf{v}^1, \ldots, \mathbf{v}^m\})$.*
3. *There exists $0 \leq m \leq d$ and orthonormal vectors $\mathbf{v}^1, \ldots, \mathbf{v}^m \in X$ such that, $X = \text{span}(\{\mathbf{v}^1, \ldots, \mathbf{v}^m\})$.*
4. *There exists $0 \leq m \leq d$ and a matrix $A \in \mathbb{R}^{(d-m)\times d}$ with full row rank such that $X = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} = \mathbf{0}\}$.*

Definition 1.2.11 The number m showing up in items 2., 3., and 4. in Theorem 1.2.10 is called the *dimension* of X. The set of vectors $\{\mathbf{v}^1, \ldots, \mathbf{v}^m\}$ are called a *basis* for the linear subspace.

1.2.1 Singular Value and Eigen Decompositions

Theorem 1.2.12 (Singular Value Decomposition (SVD)) *Let $A \in \mathbb{R}^{m\times d}$ with $\text{rk}(A) = r$. There exist scalars $\sigma_1, \ldots, \sigma_r > 0$, and orthonormal sets of vectors $\mathbf{v}^1, \ldots, \mathbf{v}^r \in \mathbb{R}^d$ and $\mathbf{u}^1, \ldots, \mathbf{u}^r \in \mathbb{R}^m$ such that*

$$A\mathbf{x} = \sigma_1 \langle \mathbf{v}^1, \mathbf{x}\rangle \mathbf{u}^1 + \cdots \sigma_r \langle \mathbf{v}^r, \mathbf{x}\rangle \mathbf{u}^r.$$

In other words, $A = U\Sigma V^T$, where Σ is an $r \times r$ diagonal matrix with $\sigma_1, \ldots, \sigma_r$ on the main diagonal, and V and U are the matrices with $\mathbf{v}^1, \ldots, \mathbf{v}^r$ and $\mathbf{u}^1, \ldots, \mathbf{u}^r$ as columns, respectively. Such a factorization of the matrix is known as a singular value decomposition *(SVD). $\sigma_1, \ldots, \sigma_r$ are called the* singular values*, $\mathbf{v}^1, \ldots, \mathbf{v}^r$ are called* right singular vectors*, and $\mathbf{u}^1, \ldots, \mathbf{u}^r$ are called* left singular vectors*.*

Up to permutations, $\sigma_1, \ldots, \sigma_r$ are unique, i.e., any SVD of A has the same Σ up to permutations of the diagonal entries. However, there may exist two distinct SVDs $U\Sigma V^T$ and $U'\Sigma V'^T$ such that U and U' do not have the same set of columns and/or V and V' do not have the same set of columns.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

Definition 1.2.13 A square matrix $A \in \mathbb{R}^{d\times d}$ is called *symmetric* if $A_{ij} = A_{ji}$ for all $i, j \in \{1, \ldots, d\}$, i.e., $A = A^T$.

Definition 1.2.14 Let $A \in \mathbb{R}^{d\times d}$. A vector $\mathbf{v} \in \mathbb{R}^d$ is called an *eigenvector* of A if there exists $\lambda \in \mathbb{R}$ such that $A\mathbf{v} = \lambda\mathbf{v}$. λ is called *the eigenvalue* of A associated with $\mathbf{v}$. $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ will denote the maximum and minimum eigenvalues of A, respectively.

Theorem 1.2.15 *If $A \in \mathbb{R}^{d\times d}$ is symmetric then it has d orthogonal eigenvectors $\mathbf{v}^1, \ldots, \mathbf{v}^d$ all of unit Euclidean norm, with associated eigenvalues $\lambda_1, \ldots, \lambda_d \in \mathbb{R}$. Moreover, if S is the matrix whose columns are $\mathbf{v}^1, \ldots, \mathbf{v}^d$ and Λ is the diagonal matrix with $\lambda_1, \ldots, \lambda_d$ as the diagonal entries, then $A = S\Lambda S^T$.*

Moreover, $\mathrm{rk}(A)$ *equals the number of nonzero eigenvalues.*

Theorem 1.2.16 *Let $A \in \mathbb{R}^{d\times d}$ be a symmetric matrix of rank r. The following are equivalent.*

1. *All eigenvalues of A are nonnegative.*
2. *There exists a matrix $B \in \mathbb{R}^{r\times d}$ with linearly independent rows such that $A = B^T B$.*
3. $\mathbf{u}^T A\mathbf{u} \geq 0$ *for all* $\mathbf{u} \in \mathbb{R}^d$.

Definition 1.2.17 A symmetric matrix $A \in \mathbb{R}^{d\times d}$ satisfying any of the three conditions in Theorem 1.2.16 is called a *positive semidefinite* (PSD) matrix. If $\mathrm{rk}(A) = d$, i.e., all its eigenvalues are strictly positive, then A is called *positive definite*. Any matrix B satisfying condition 2. in Theorem 1.2.16 is called a *square root* of A.

The following relationship between the SVD of a matrix A and the eigenvectors and eigenvalues of AA^T and $A^T A$ is useful.

Proposition 1.2.18 *Let $A \in \mathbb{R}^{m\times d}$. The following are true.*

1. *σ is a singular value of A if and only if it is the square root of a positive eigenvalue of AA^T and $A^T A$. In particular, for positive semidefinite matrices, singular values and eigenvalues coincide.*
2. $\mathbf{u}$ *is a left singular vector of A if and only if it is an eigenvector of AA^T.*
3. $\mathbf{v}$ *is a right singular vector of A if and only if it is an eigenvector of $A^T A$.*

1.2.2 Matrix Norms

It is sometimes useful to consider the space $\mathbb{R}^{m\times d}$ matrices as a Euclidean space in its own right. This can be done explicitly by "vectorizing" the matrix,

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

i.e., thinking of $A \in \mathbb{R}^{m\times d}$ as a vector in $\mathbb{R}^{md}$. There are several ways one can do this, depending on how the matrix entries are ordered as coordinates of the corresponding vector. This choice of ordering will not matter at all for what follows. What is important is that the operations of matrix addition and multiplication by a scalar coincide with the standard vector space structure on the corresponding Euclidean space. The notion of a norm and the distance induced by this norm are then well-defined concepts on $\mathbb{R}^{m\times d}$ (or $\mathbb{R}^{md}$). The following particular class of norms will be useful.

Definition 1.2.19 Let N be a norm on $\mathbb{R}^d$ and N' be a norm on $\mathbb{R}^m$. The *induced norm* on $\mathbb{R}^{m\times d}$ is defined as

$$\|A\|_{N,N'} := \sup_{\mathbf{x}\in\mathbb{R}^d\setminus\{\mathbf{0}\}} \frac{N'(A\mathbf{x})}{N(\mathbf{x})}.$$

When $N = \ell_p$ and $N' = \ell_q$ for some $p, q \geq 1$, this will be abbreviated to $\|A\|_{p,q}$, and when $p = q$, to simply $\|A\|_p$. The special case of $p = q = 2$ has the name *spectral norm* of A.

The following is a straightforward consequence of the definitions.

Theorem 1.2.20 *The spectral norm of A is given by the largest singular value of A.*

1.2.3 Exercises

1. Show that any positive definite matrix $A \in \mathbb{R}^{d\times d}$ defines a norm on $\mathbb{R}^d$ via $N_A(\mathbf{x}) = \sqrt{\mathbf{x}^T A\mathbf{x}}$. This norm is called the *norm induced by A*. (When A is the identity matrix I_d, this gives the standard Euclidean norm.)
2. Let $A \in \mathbb{R}^{d\times d}$ be a positive definite matrix. Show the generalized Cauchy–Schwarz inequalities: $|\mathbf{y}^T A\mathbf{x}| \leq N_A(\mathbf{x}) \cdot N_A(\mathbf{y})$ and $|\mathbf{y}^T\mathbf{x}| \leq N_A(\mathbf{x}) \cdot N_{A^{-1}}(\mathbf{y})$.
3. Show that a function $T : \mathbb{R}^d \to \mathbb{R}^m$ is an affine transformation if and only if $T(\lambda\mathbf{x} + \gamma\mathbf{y}) = \lambda T(\mathbf{x}) + \gamma T(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and all $\lambda, \gamma \in \mathbb{R}$ such that $\lambda + \gamma = 1$.
4. Let $A \in \mathbb{R}^{k\times k}$. Show that for any $\mathbf{x} \in \mathbb{R}^k$,

$$\sigma_{\min}(A)\|\mathbf{x}\|_2 \leq \|A\mathbf{x}\|_2 \leq \sigma_{\max}(A)\|\mathbf{x}\|_2,$$

where $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ are the smallest and largest singular values of the matrix A, respectively, with the convention that if A is not invertible then we take the smallest singular value to be 0.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

1.3 Real Analysis

For any subset of real numbers $S \subseteq \mathbb{R}$, we denote the *infimum* by $\inf S$ and the *supremum* by $\sup S$. The following are useful properties of infimums and supremums (we use the set operations from Definition 1.1.8 applied to $\mathbb{R}^d$ with $d = 1$).

Theorem 1.3.1 *Let $A, B \subseteq \mathbb{R}$ and $t \geq 0$. The following are all true.*

1. $\inf(tA) = t\inf(A)$.
2. $\inf(A + B) = \inf(A) + \inf(B)$.
3. *If $A \subseteq B$ then* $\inf(A) \geq \inf(B)$.
4. $\inf(-A) = -\sup(A)$, *where* $-A = \{-x : x \in A\}$.

Definition 1.3.2 Fix a norm N on $\mathbb{R}^d$. A set $X \subseteq \mathbb{R}^d$ is called *open (with respect to N)* if for every $\mathbf{x} \in X$, there exists $r \in \mathbb{R}_+$ such that $B_N(\mathbf{x}, r) \subseteq X$. A set X is *closed (with respect to N)* if its complement $\mathbb{R}^d \setminus X$ is open.

By Theorem 1.1.3, a set $X \subseteq \mathbb{R}^d$ is open with respect to a norm N if and only if X is open with respect to every norm on $\mathbb{R}^d$. Thus, we can drop the qualification "with respect to a norm" for referring to open and closed sets.

Theorem 1.3.3 *The following all hold:*

1. $\emptyset, \mathbb{R}^d$ *are both open and closed.*
2. *An arbitrary union of open sets is open. An arbitrary intersection of closed sets is closed.*
3. *A finite intersection of open sets is open. A finite union of closed sets is closed.*

Definition 1.3.4 A *sequence* in $\mathbb{R}^d$ is a countable ordered set of points, namely $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots$, and will often be denoted by $\{\mathbf{x}^i\}_{i \in \mathbb{N}}$. We say that *the sequence converges* or that *the limit of the sequence exists* if there exists a point $\mathbf{x}$ such that for every $\epsilon > 0$, there exists $M \in \mathbb{N}$ such that $N(\mathbf{x} - \mathbf{x}^n) \leq \epsilon$ for all $n \geq M$, for some norm N on $\mathbb{R}^d$. $\mathbf{x}$ is called the *limit point*, or simply the *limit*, of the sequence and will also sometimes be denoted by $\lim_{n \to \infty} \mathbf{x}^n$.

By Theorem 1.1.3, the concept of a limit does not depend on the choice of the norm: $\mathbf{x}$ is the limit of a sequence $\{\mathbf{x}^i\}_{i \in \mathbb{N}}$ under some norm if and only if it is the limit under every norm.

Theorem 1.3.5 *A set $X \subseteq \mathbb{R}^d$ is closed if and only if for every convergent sequence in X, the limit of the sequence is also in X.*

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

Definition 1.3.6 We introduce three important notions:

1. For any set $X \subseteq \mathbb{R}^d$, the *closure of* X is the smallest (with respect to set inclusion) closed set containing X and will be denoted by $\mathrm{cl}(X)$.
2. For any set $X \subseteq \mathbb{R}^d$, the *interior of* X is the largest (with respect to set inclusion) open set contained inside X and will be denoted by $\mathrm{int}(X)$.
3. For any set $X \subseteq \mathbb{R}^d$, the *boundary of* X is defined as $\mathrm{bd}(X) := \mathrm{cl}(X) \setminus \mathrm{int}(X)$.

Definition 1.3.7 Let $X' \subseteq X \subseteq \mathbb{R}^d$ be two subsets. X' is said to be *dense in* X if $X \subseteq \mathrm{cl}(X')$.

Definition 1.3.8 A set in $\mathbb{R}^d$ that is closed and bounded is called *compact.*

Theorem 1.3.9 (Heine–Borel theorem) *Let $C \subseteq \mathbb{R}^d$ be a compact set, and let $\{U_\lambda : \lambda \in \Lambda\}$ be any (possibly infinite) family of open subsets of $\mathbb{R}^d$ such that $C \subseteq \bigcup_{\lambda\in\Lambda} U_\lambda$. Then there exists a finite subfamily $\Lambda' \subseteq \Lambda$ such that $C \subseteq \bigcup_{\lambda\in\Lambda'} U_\lambda$.*

Theorem 1.3.10 *Let $C \subseteq \mathbb{R}^d$ be a compact set. Then every sequence $\{\mathbf{x}^i\}_{i\in\mathbb{N}}$ contained in C (not necessarily convergent) has a convergent subsequence.*

Theorem 1.3.11 *Let C_λ, $\lambda \in \Lambda$ be any family of closed sets in $\mathbb{R}^d$ such that at least one of them is compact. $\bigcap_{\lambda\in\Lambda} C_\lambda = \emptyset$ if and only if there is a finite set of indices $\lambda_1, \ldots, \lambda_k \in \Lambda$ such that $C_{\lambda_1} \cap \cdots \cap C_{\lambda_k} = \emptyset$.*

Definition 1.3.12 A function $f\colon \mathbb{R}^d \to \mathbb{R}^n$ is *continuous at* $\mathbf{x} \in \mathbb{R}^d$ if for every convergent sequence $\{\mathbf{x}^i\}_{i\in\mathbb{N}} \subseteq \mathbb{R}^d$ with $\lim_{n\to\infty} \mathbf{x}^i = \mathbf{x}$, we have $\lim_{i\to\infty} f(\mathbf{x}^i) = f(\mathbf{x})$. A function is said to be *continuous* if it is continuous at every $\mathbf{x} \in \mathbb{R}^d$.

Let N_1 be a norm on $\mathbb{R}^d$ and N_2 be a norm on $\mathbb{R}^n$. f is said to be *Lipschitz continuous (with respect to these norms) over a domain $D \subseteq \mathbb{R}^d$* if there exists a constant L such that $d_{N_2}(f(\mathbf{x}), f(\mathbf{y})) \leq L \cdot d_{N_1}(\mathbf{x}, \mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in D$. L is called the *Lipschitz constant of f over D with respect to these norms*.

It can be verified that a Lipschitz continuous function is continuous.

Theorem 1.3.13 (Weierstrass' theorem) *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a continuous function. Let $X \subseteq \mathbb{R}^d$ be a nonempty, compact subset. Then* $\inf\{f(\mathbf{x}) : \mathbf{x} \in X\}$ *is attained, i.e., there exists $\mathbf{x}^{\min} \in X$ such that $f(\mathbf{x}^{\min}) = \inf\{f(\mathbf{x}) : \mathbf{x} \in X\}$. Similarly, there exists $\mathbf{x}^{\max} \in X$ such that $f(\mathbf{x}^{\max}) = \sup\{f(\mathbf{x}) : \mathbf{x} \in X\}$.*

A generalization of the above theorem is the following.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

Theorem 1.3.14 *Let $f: \mathbb{R}^d \to \mathbb{R}^n$ be a continuous function, and C be a compact set. Then $f(C)$ is compact.*

We will also need to speak of differentiability of functions $f: \mathbb{R}^d \to \mathbb{R}^n$.

Definition 1.3.15 We say that $f: \mathbb{R}^d \to \mathbb{R}^n$ is differentiable at $\mathbf{x} \in \mathbb{R}^d$ if there exists a linear transformation $A: \mathbb{R}^d \to \mathbb{R}^n$ such that

$$\lim_{\mathbf{h}\to\mathbf{0}} \frac{\|f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}) - A\mathbf{h}\|}{\|\mathbf{h}\|} = 0.$$

If f is differentiable at $\mathbf{x}$, then the linear transformation A is unique. It is commonly called the *differential* or *total derivative of f at* $\mathbf{x}$ and is denoted by $f'(\mathbf{x})$. When $n = 1$, $f'(\mathbf{x})$ is a linear functional and can therefore be represented by an inner product, i.e., there exists a vector $\mathbf{v} \in \mathbb{R}^d$ such that $f'(\mathbf{x})(\mathbf{u}) = \langle \mathbf{v}, \mathbf{u} \rangle$ for all $\mathbf{u} \in \mathbb{R}^d$. The vector $\mathbf{v}$ is commonly called the *gradient of f* and is denoted by $\nabla f(\mathbf{x})$.

If $f: \mathbb{R}^d \to \mathbb{R}$ is differentiable everywhere, and the gradient function $\nabla f: \mathbb{R}^d \to \mathbb{R}^d$ is differentiable at $\mathbf{x}$, then its differential at $\mathbf{x}$ is a linear map from $\mathbb{R}^d$ to $\mathbb{R}^d$. The corresponding matrix (see Section 1.2) is called the *Hessian of f at* $\mathbf{x}$, and it is denoted by $\nabla^2 f(\mathbf{x})$. f is said to be *twice differentiable* if ∇f is differentiable everywhere.

Definition 1.3.16 Let $f: \mathbb{R}^d \to \mathbb{R}$ be any function and let $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$. We define the *directional derivative of f at* $\mathbf{x}$ *in the direction* $\mathbf{r}$ as

$$f'(\mathbf{x}; \mathbf{r}) := \lim_{t\to 0^+} \frac{f(\mathbf{x}+t\mathbf{r}) - f(\mathbf{x})}{t} \tag{1.3.1}$$

if that limit exists. Note that we consider the limit as t approaches 0 from the right. We will be speaking of $f'(\mathbf{x}; \cdot)$ as a function from $\mathbb{R}^d \to \mathbb{R}$.

For any coordinate $i \in \{1, \ldots, d\}$, if $f'(\mathbf{x}; \mathbf{e}^i) = -f'(\mathbf{x}, -\mathbf{e}^i)$ then the limit

$$\lim_{h\to 0} \frac{f(\mathbf{x}+h\mathbf{e}^i) - f(\mathbf{x})}{h}$$

exists and is called the *partial derivative of $f: \mathbb{R}^d \to \mathbb{R}$ at* $\mathbf{x}$ *in the ith direction*. It will be denoted by $f_i'(\mathbf{x})$.

Theorem 1.3.17 *If $f: \mathbb{R}^d \to \mathbb{R}$ is differentiable at $\mathbf{x}$, then the partial derivatives exist at $\mathbf{x}$ for all $i = 1, \ldots, d$ and the i-coordinate of $\nabla f(\mathbf{x})$ is precisely $f_i'(\mathbf{x})$. Conversely, if the partial derivatives all exist at $\mathbf{x}$ and they are continuous functions at $\mathbf{x}$, then f is differentiable at $\mathbf{x}$.*

Definition 1.3.18 For any subset $X \subseteq \mathbb{R}^d$, $\mathrm{vol}(X)$ will denote the *volume* of X, i.e., $\mathrm{vol}(X) := \int_X d\mathbf{x}$.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

The following limit property of volumes will be used.

Theorem 1.3.19 *Let $X_1, X_2, \ldots$ be a sequence of sets in $\mathbb{R}^d$ such that $X_i \subseteq X_j$ for all $i \leq j$. Then* $\mathrm{vol}\left(\bigcup_{i=1}^{\infty} X_i\right) = \lim_{i\to\infty} \mathrm{vol}(X_i)$.

Similarly, if $X_1, X_2, \ldots$ be a sequence of compact sets in $\mathbb{R}^d$ such that $X_i \supseteq X_j$ for all $i \leq j$. Then $\mathrm{vol}\left(\bigcap_{i=1}^{\infty} X_i\right) = \lim_{i\to\infty} \mathrm{vol}(X_i)$.

1.3.1 Exercises

1. Show that if X is compact and Y is closed, then $X + Y$ is closed. Find an example with closed sets X, Y such that $X + Y$ is not closed.
2. Show that if f is differentiable everywhere, it is also continuous everywhere.

1.4 Models of Computation

The second part of this book deals with mathematical optimization, with a heavy emphasis on algorithmic aspects. This necessitates a discussion of a model of computation in which we will carry out all the operations for solving an optimization problem.

We will not enter into a very detailed and formal discussion of computation models. At an intuitive level, an algorithm is simply a piece of computer code that takes as input an instance of an optimization problem and outputs the solution to it. However, there are two related issues that require clarification. First, one has to formalize what it means to "input an instance of an optimization problem." For example, we will discuss at length optimization using functions defined over $\mathbb{R}^d$ and subsets of $\mathbb{R}^d$. We have to define more precisely how we can "input" a function or a subset to a computer. Second, since we will be dealing with functions over $\mathbb{R}^d$ and subsets of $\mathbb{R}^d$, we may have to deal with irrational numbers in $\mathbb{R}$. How are we to represent arbitrary real numbers on a "finite memory" computer?

These two issues bring us face-to-face with two distinct approaches to computation within mathematical optimization. The *continuous* or *numerical* optimization community, with its very close connection to scientific computation and numerical analysis, typically works in the *arithmetic model* of computation. On the other hand, the *discrete* or *combinatorial* optimization community, due to a more significant overlap with computer science, primarily uses the *Turing machine model* of computation. We discuss both of these models in a little more detail below.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

1.4.1 Arithmetic Model of Computation

Here one assumes that one can perform addition, subtraction, multiplication, and division of any two real numbers, as well as compare two real numbers to tell whether the first one is lesser than, greater than, or equal to the second number. In some situations, one also allows the possibility of taking square roots. These operations are called *elementary operations*. Thus, this model of computation is over the ordered field of real numbers: arbitrary real numbers can be given as input, be stored in memory, and be outputs. An algorithm in this model of computation is a piece of computer code (with the standard notions of programming variables, iteration loops, branchings, etc.) with the idealization just mentioned: one can store arbitrary real numbers in memory and perform computations on them using the elementary operations. Different algorithms will have different formats for their inputs, but in the end it is simply a list of real numbers that encodes the input.

Given an algorithm in this model, and any input to the algorithm, the *running time* or *time complexity* of the algorithm on this input is the total number of elementary operations performed by the algorithm. Similarly, the *space complexity* of the algorithm is the maximum amount of memory used (i.e., the number of real numbers stored) at any given point during the algorithm's execution.

1.4.2 Turing Machine Model of Computation

Here, one breaks down the notion of computation to even more basic components. For instance, the addition of two numbers is not an elementary operation: intuitively, more computation is needed to add larger numbers compared to smaller numbers. The main difference from the arithmetic model is that everything – inputs, outputs, and intermediate objects in memory – must be stored as finite-length binary strings and one can perform three simple operations: read a bit stored in a particular location in memory (and check if it equals 0 or 1), change the bit stored in a particular location in memory, or write a bit into a new memory location. There are many equivalent ways of formalizing what an algorithm is in this model, which we will not get into here; see the references at the end of the chapter. It suffices for our purposes to think of an algorithm in this model of computation as a piece of computer code (with the standard notions of programming variables, iteration loops, branchings, etc.) with elementary operations restricted to the above operations involving bits.

Given an algorithm in this model, and any input to the algorithm, the *running time* or *time complexity* of the algorithm on this input is again the

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

total number of elementary operations performed by the algorithm. Similarly, the *space complexity* of the algorithm is the maximum amount of memory used (i.e., the number of bits stored) at any given point during the algorithm's execution.

In our opinion, both of these approaches have their advantages and disadvantages and no one approach captures all the subtleties of what it means to solve an optimization problem. Discrete or combinatorial optimization has traditionally been concerned with objects that have a natural encoding as finite binary strings, and so the Turing machine model of computation is a natural way to analyze such algorithms. However, continuous optimization, by its very nature, deals with the ordered set of reals in its computations and the arithmetic model becomes the obvious model. The Turing machine model, with its finitary nature, cannot (exactly) represent the set of all real numbers. From a mathematical perspective, one could argue that the arithmetic model is more general, since it can simulate any computation performed in the Turing machine model, since finite binary strings are simply a list of 0's and 1's, and the elementary operations of a Turing machine can be simulated by the elementary operations of the arithmetic model. In fact, under standard computational complexity assumptions and depending on exactly which programming instructions are allowed, it can be shown rigorously that the arithmetic model is significantly more powerful than the Turing machine model; see Section 1.5 for more on the comparisons between these two models of computation. However, currently, no real (physical) computing machine can implement the arithmetic model. Thus, one could counterargue that the Turing machine model is the one that can actually be implemented and thus more authentic as a mathematical model of computation. We personally think this counterargument is a bit narrow. First, we cannot predict what physical machines may or may not be invented in the future; perhaps, some analog machines will be able to implement the arithmetic model. Second, valuable mathematical insights into numerical optimization can be derived from the arithmetic model of computation, which is impossible to express in the Turing machine model.

We do not mean to enter into a deep philosophical debate on the possibilities and limitations of different computational models. This inevitably leads to foundational questions in mathematics (e.g., the nature of existence of an irrational number) and computation (e.g., designing an algorithm for adding arbitrary real numbers) which is not the focus of this book. Instead, we will adopt a third approach, outlined in Section 1.4.3, that interpolates between these two models and is very convenient for discussing optimization algorithms.

Before we proceed, we point out that Turing machines can perform computations over rational numbers using a standardized encoding which represents any integer using its binary representation and a rational number as a pair of (binary representations of) integers (numerator and denominator). The *encoding size* of any rational number is the size of the binary string used to represent it. We generalize this notion to both models of computation.

Definition 1.4.1 Let X be an arbitrary set. We say that X *has a representation in the arithmetic model of computation* if there is an injective map from X to the set of finite sequences of real numbers. Similarly, we say that X *has a representation in the Turing machine model of computation* if there is an injective map from X to the set of finite binary strings. These maps give "labels" or *encodings* for the elements of X so that they can be stored in memory if needed and an algorithm can process them in its computations. The *encoding size* of any $x \in X$ is the length of the sequence that encodes it.

1.4.3 Oracle-Based Computation

1.4.3.1 Turing Machines Augmented with Real Number Oracles

The tension noted above between the two models led to a hybrid approach that tries to retain the richness of the arithmetic model while staying close to the principle of physical realizability of the Turing machine model. It traces its roots to the constructive philosophy of mathematics [40, 46], and considers the Turing machine model augmented with *oracles*. What this means is that an algorithm can query certain oracles at any point during its execution and use the responses in its computations. The responses from the oracle must be binary strings so that they can be processed by the Turing machine. The most basic kind of an oracle is that representing a set of real numbers.

Definition 1.4.2 Let $\mathcal{I} \subseteq \mathbb{R}$ be a subset of real numbers. A *rational oracle* representing $\mathcal{I}$ is a set of functions $\{q_\epsilon\}_{\epsilon \in \mathbb{Q}_+ \setminus \{0\}}$ indexed by the positive rationals with $q_\epsilon : \mathcal{I} \to \mathbb{Q}$ such that for any $\alpha \in \mathcal{I}$, $|q_\epsilon(\alpha) - \alpha| \leq \epsilon$.

Real number oracles are sometimes equipped with a "size": for every $\alpha \in \mathcal{I}$, there exists a constant $K_\alpha \geq 1$ such that for every $\epsilon > 0$, the encoding size of the rational number q_ϵ is at most K_α times the encoding size of ϵ. Additionally if α is rational, K_α must be at least the encoding size of α. This ensures that any algorithm in the standard Turing machine model has a comparable implementation in the oracle-based model. See [168, Section 1.4] for this important point.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

1.4.3.2 General Oracles

In Part II of this book, we will encounter other oracles, e.g., those representing functions on $\mathbb{R}^d$ and subsets of $\mathbb{R}^d$, that generalize the basic real number oracles from Definition 1.4.2. This will also tie into a formalization of the idea of giving an optimization problem as "input" to an algorithm. In this context, it will be useful to extend the notion of oracle-based computation to the arithmetic model as well.

Definition 1.4.3 Let $\mathcal{I}$ be an arbitrary set. An *oracle representing* $\mathcal{I}$ is given by a set $\mathcal{Q}$ of possible *queries* and a set H of possible *answers* or *responses*. Each query $q \in \mathcal{Q}$ is a function $q\colon \mathcal{I} \to H$. We say that $q(I) \in H$ is the answer (or response) to the query q on the element $I \in \mathcal{I}$. The oracle is said to be *unambiguous* if for every $I, I' \in \mathcal{I}$ with $I \neq I'$, there exists some $q \in \mathcal{Q}$ such that $q(I) \neq q(I')$.

An oracle is *compatible with a model of computation* (arithmetic or Turing machine) if $\mathcal{Q}$ and H both have representations in that model (Definition 1.4.1).

An *oracle-based algorithm for processing* $\mathcal{I}$ in either of these two models of computation that uses an oracle $(\mathcal{Q}, H)$ compatible with that model of computation is an algorithm that has the additional ability to query any $q \in \mathcal{Q}$ and use the response in its computations. Such algorithms do not have an explicit input $I \in \mathcal{I}$, but rather the input is revealed implicitly by the responses $q(I)$ it receives from the queries to the oracle.

In the traditional view of computation the set of possible inputs $\mathcal{I}$ to the algorithm has an encoding itself, i.e., $\mathcal{I}$ has a representation in the model of computation (Definition 1.4.1). One can then associate a natural unambiguous oracle compatible with this model of computation: $\mathcal{Q} = \{q\}$ is a singleton and $q(I)$ is the encoding of $I \in \mathcal{I}$. Algorithms processing $\mathcal{I}$ can then simply query q at the beginning of their computation (equivalent to "receiving the input") and then never make any other oracle queries. Therefore, oracle-based computation strictly expands the scope of computation from the traditional view of "inputs processed to give outputs." In fact, adaptively posing the queries will be a crucial feature of several mathematical optimization algorithms we discuss in Part II. We will see concrete examples of oracles for different kinds of optimization problems (cf. Example 5.2.2 and Section 6.1.1).

The arithmetic model often leads to the most concise and elegant description of algorithmic ideas in optimization and we will mostly use this model in our discussions. Moreover, for some of the algorithms we discuss it is not known if a version in the Turing machine model (with or without oracles) exists. Nevertheless, we will provide references to implementations

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press

in the (oracle) Turing machine model for all the algorithms where such implementations are known in the literature.

1.5 Notes and Related Literature

Euclidean spaces and their linear algebraic and real analytic properties are well-studied topics, with many books covering them that fill several shelves in an academic library. We recommend [219] for the linear algebra of Euclidean spaces. Halmos' text [132] is a superb reference for a "coordinate-free" study of finite-dimensional vector spaces and linear algebra. Two classic references for real analysis (both in Euclidean spaces and more generally) are [204, 206]. These three books cover everything that we surveyed in Sections 1.1, 1.2, and 1.3, and much more.

The Turing machine model of computation has been studied for almost a century now. A good introduction is [5, chapter 1], which includes a discussion of related models like *(integer) random access machines (RAMs)*, and [13] dives into the subject in depth.

The arithmetic model has its roots in attempts to formalize numerical algorithms such as Newton's method for finding roots of nonlinear equations. It seems hard to formulate such classical computational procedures in numerical analysis and scientific computation in the Turing machine model, simply because such procedures assume operations over the entire ordered field of real numbers which is impossible to capture in the finitary world of the Turing machine model. Several proposals have been put forward to address this issue; see [1, 53, 56, 118, 123, 129, 155, 158, 168, 189, 191, 224, 230] as a representative list. The arithmetic model, as described in Section 1.4, is essentially the same as the computational models from [56, 191], very closely related to the model in [53], and is also sometimes referred to as the *real RAM* model. As mentioned in Section 1.4.2, it has been shown rigorously that the real RAM model is significantly more powerful, unless one restricts the programming instructions or arithmetic operations [43, 133, 153, 190, 211]. For instance, it can be shown that if something called *indirect indexing* (an operation which is present in every modern programming language) is allowed in the real RAM model, it can solve in polynomial time all problems solvable by Turing machines with polynomial amount of memory. This represents a significant increase in power because if polynomial time is the same as polynomial space for Turing machines, the so-called *polynomial hierarchy* would collapse, and among other things, the class P would equal NP [13]. See [115] for some recent work on the real RAM model that addresses this issue.

The oracle Turing machine model has a long history going back to the original work of Turing [224, 225], and developed further in [129, 158]. We refer to [155, 230] as good textbook expositions. Using this model in the realm of (discrete) optimization seems to have been first explored in [123, 168]. In particular, Definition 1.4.2 is taken directly from [168].

Our definition of a general oracle and oracle-based algorithms in this general setting is not explicitly stated anywhere in the literature. However, these concepts and their use in Part II are very much inspired by the beautiful monograph [222], as well as work in the oracle Turing machine model cited above. Definitions 1.4.1 and 1.4.3 are made with respect to the arithmetic model and the Turing machine model of computation, but they can be easily adapted to any other model of computation, including the ones cited above. This makes the view presented here of algorithmic computation very flexible and general, as well as especially useful for discussing diverse kinds of mathematical optimization algorithms (combinatorial and numerical) under a unifying umbrella in Part II.

https://doi.org/10.1017/9781108946650.002 Published online by Cambridge University Press