Massimiliano de Sa
Haeyoon Han

Lectures on Linear Systems Theory

Draft: January 2025

Caltech Control & Dynamical Systems

Preface

These lecture notes on linear systems theory are designed for a roughly 10 week course on the subject. The aim is to provide a balanced overview of both the state space and I/O perspectives on linear systems, with an eye towards precise mathematical formulations of problems in control. The approach we follow is inspired by the Fall 2023 offering of CDS 131, Linear Systems Theory, by John Doyle.

Texts that have notably influenced our presentation of the material include *Linear System Theory* by Callier and Desoer, *Mathematical Control Theory* by Sontag, and *Feedback Control Theory* by Doyle, Francis, and Tannenbaum. The lecture notes on linear systems theory by John Lygeros and those by Richard Murray have also proven to be invaluable resources, as has as Stephen Boyd's course EE 363, Linear Dynamical Systems.

Each section of the text corresponds to a roughly 1.5 hour lecture. Subsections that can be skipped without loss of continuity are marked with a ★. To demarcate the difficulty of problems, we also use a star system; ★ means challenging and ★★ means very challenging, compared to the average unstarred problem.

Although the only formal prerequisites for this course are a strong background in linear algebra, prior exposure to control systems and real analysis is certainly useful. Typically, PhD students in control taking this course have either taken or are concurrently taking courses in convex optimization and linear functional analysis, and have had a first (undergraduate) course in control. Prior knowledge of these subjects is not, however, required in order to succeed in learning the course material. We provide a cursory review of the essentials in Chapter 1.

Contents

Chapter 1
Mathematical Preliminaries

As its name suggests, control *theory* is a fundamentally mathematical subject. At the start of a course in linear systems theory, one is often expected to have mastered topics from linear algebra, real analysis, and functional analysis. Given the modern day engineering curriculum, it's not altogether realistic to expect students to know all of these right from the get-go.

In this brief chapter, we introduce the fundamental mathematical concepts needed to study mathematical systems theory. We stress - you *don't* need to master everything in this chapter upon the first read! If it's your first time seeing this material, try to get a basic feel for the definitions in your first read - you can always come back and refresh your knowledge as the course progresses.

Math is a contact sport, and learning to deal with mathematical abstraction is something that only comes with practice. As such, we leave many of the easier results of this chapter as exercises - you're encouraged to do them as you go along to build your understanding. With this said, let's begin!

1.1 Vector Spaces

The fundamental setting for linear system theory is the *vector space*. In this section, we'll get to grips with the basic definitions and properties of vector spaces. Notably, we'll construct very general definitions of vector spaces which don't assume finite-dimensionality.

First, let's construct a formal definition of a vector space. From your first course in linear algebra, you might recall working in vector spaces such as $\mathbb{R}^n$ or $\mathbb{C}^n$, in which *vectors* are tuples of real or complex numbers. You might have defined *vector addition* as the operation which takes two tuples of numbers and adds each pair of entries to form a new vector, or *scalar multiplication*, which multiplies each element of the vector by a scalar to form a new vector. The following definition of a vector space takes the fundamental properties of vectors, addition, and scalar multiplication that are familiar to us from $\mathbb{R}^n$ and $\mathbb{C}^n$, and abstracts away the "tuples of numbers" into an abstract vector.

Definition 1.1 (Vector Space) Let $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. A vector space V over $\mathbb{K}$ is a set, V, together with two operations, $+ : V \times V \to V$ and $(\cdot) : \mathbb{K} \times V \to V$, satisfying:

1. Closure under operations: For all $u, v \in V$ and all $\alpha, \beta \in \mathbb{K}$, $\alpha u + \beta v \in V$.

2. Associativity: For all $u, v, w \in V$, $u + (v + w) = (u + v) + w$.
3. Commutativity: For all $u, v \in V$, $u + v = v + u$.
4. Additive Identity: There exists an element $0 \in V$ for which $0 + v = v$, for all $v \in V$.
5. Additive Inverse: For all $v \in V$, there exists an element $-v \in V$ for which $v + (-v) = 0$.
6. Compatibility: For all $\alpha, \beta \in \mathbb{K}$ and $v \in V$, $\alpha \cdot (\beta \cdot v) = (\alpha\beta) \cdot v$.
7. Multiplicative Identity: For all $v \in V$, $1 \cdot v = v$.
8. Distributivity in V: For all $\alpha \in \mathbb{K}$ and $u, v \in V$, $\alpha \cdot (u + v) = \alpha u + \alpha v$.
9. Distributivity in $\mathbb{K}$: For all $\alpha, \beta \in \mathbb{K}$ and $v \in V$, $(\alpha + \beta) \cdot v = \alpha v + \beta v$.

V is called the *set of vectors* and $\mathbb{K}$ is called the *field*. Elements of the set V are called *vectors*, while elements of the set $\mathbb{K}$ are called *scalars*.

Remark 1.1 Above, we used $(\cdot)$ to denote the multiplication of a vector by a scalar. One typically suppresses the $(\cdot)$, and writes $c \cdot v = cv$, for $c \in \mathbb{K}$ and $v \in V$.

Remark 1.2 Formally, we denote a vector space V over $\mathbb{K}$ by the pair $(V, \mathbb{K})$. However, if $\mathbb{K}$ is clear from context, one refers to a vector space simply by the set V.

Let's take a moment to appreciate what's going on underneath all of the abstraction. First, let's take a moment to outline the structure of an abstract definition, for those who might be unfamiliar with definition-theorem-proof mathematics. Typically, when writing an abstract definition, one starts by specifying out a couple of objects - here, we specify a set, V, a set $\mathbb{K}$, and the operations $+$ and $(\cdot)$. Then, we specify some *axioms* - properties that the objects must have. By laying out each definition in this manner, we can ensure there are no ambiguities in the foundations of our theory.

With this in mind, let's think about what Definition 1.1 is actually saying. All that Definition 1.1 *really* says is that a vector space is any set V along with a set of scalars $\mathbb{K}$, equipped with two operations $+$ and $(\cdot)$ that act like $+$ and $(\cdot)$ on $\mathbb{R}^n$ with scalars in $\mathbb{R}$. Each condition of the definition simply specifies a property that we have between tuples of numbers in $\mathbb{R}^n$, so that our *abstract* vector space behaves just like $\mathbb{R}^n$ might. The definition is summarized as follows:

1. V is a set of *vectors*, analogous to tuples of real numbers in $\mathbb{R}^n$.
2. $\mathbb{K}$ is a set of *scalars* called the field, analogous to scalar real numbers in $\mathbb{R}$.
3. The operations $+$ and $(\cdot)$ are defined to behave just like $+$ and $(\cdot)$ between real number scalars and tuples of real numbers in $\mathbb{R}^n$.

Let's consider a few examples to make things more concrete.

Example 1.1 Consider the vector space $(\mathbb{R}^n, \mathbb{R})$ (read as $\mathbb{R}^n$ over $\mathbb{R}$). Here, $V = \mathbb{R}^n$, the set of tuples of n real numbers, the set of scalars is $\mathbb{K} = \mathbb{R}$, and the operations $+$ and $(\cdot)$ are defined,

$$(x_1, ..., x_n) + (y_1, ..., y_n) = (x_1 + y_1, ..., x_n + y_n) \tag{1.1}$$

$$c \cdot (x_1, ..., x_n) = (c \cdot x_1, ..., c \cdot x_n), \tag{1.2}$$

where $c \in \mathbb{R}$ is any scalar in $\mathbb{R}$ and $(x_1, ..., x_n)$, $(y_1, ..., y_n)$ are tuples of n real numbers.

Example 1.2 Consider the vector space $(\mathbb{C}^{n\times n}, \mathbb{C})$ (read $\mathbb{C}^{n\times n}$ over $\mathbb{C}$), in which the set of vectors is $V = \mathbb{C}^{n\times n}$, the set of $n \times n$ matrices with complex entries, the set of scalars is $\mathbb{K} = \mathbb{C}$, and the operations $+$ and $(\cdot)$ are defined,

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & \dots & a_{1n}+b_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1}+b_{n1} & \dots & a_{nn}+b_{nn} \end{bmatrix} \tag{1.3}$$

$$c \cdot \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} c \cdot a_{11} & \dots & c \cdot a_{1n} \\ \vdots & \ddots & \vdots \\ c \cdot a_{n1} & \dots & c \cdot a_{nn} \end{bmatrix}. \tag{1.4}$$

Thus, we observe that the set of $n \times n$ complex matrices forms a vector space over $\mathbb{C}$.

So far, the two examples we've considered have been fairly routine. For the next example, we consider something a little bit more abstract.

Example 1.3 Let V be a set and W a vector space over $\mathbb{R}$. Consider the vector space $(\mathcal{F}_{V,W}, \mathbb{R})$, where the vectors are functions $f : V \to W$, and the operations $+, (\cdot)$ are defined by function addition and function scalar multiplication,

$$(f+g)(v) = f(v) + g(v), \;\; v \in V \tag{1.5}$$

$$(c \cdot f)(v) = c \cdot f(v), \qquad v \in V, \; c \in \mathbb{R}, \tag{1.6}$$

where in the right hand side of the expressions above, the operations $+, (\cdot)$ are those from the vector space W. Thus, we can generate *new* vector spaces from existing vector spaces! Note that we *don't require* V to be a vector space for this construction to work, only W.

This example is certainly more abstract than those we've considered so far, but follows the *same* underlying principles of a vector space. All we have is a set of vectors, a set of scalars, an addition rule, and a scalar multiplication rule. This example highlights a few important principles:

1. Vectors are more than tuples of numbers: Vectors in a vector space are *not* just tuples of numbers - they can take on far more abstract forms such as matrices and functions between vector spaces.
2. Abstraction is your friend: When thinking of vector spaces such as the space of functions above, thinking about the vectors as functions between spaces can get confusing! Instead of *looking inside* the set of vectors, use abstraction to your advantage! When performing any standard algebraic operations, you can *forget* about the complex internal structure, and abstract everything away behind the definition of a vector space. Similarly to computer science, where we hide away implementation details behind classes and methods, we hide away the "implementation" of individual vector spaces (e.g. vectors are *functions*) behind the nice abstraction of Definition 1.1. Using abstraction to your advantage is critical to managing complexity in mathematics.

Exercise 1.1 Verify that the examples presented above are indeed vector spaces by showing they satisfy all of the axioms of Definition 1.1.

Let's build upon the basic structure of a vector space we outlined above. One of the axioms of a vector space, *closure under operations*, states that for all $u, v \in V$ and $\alpha, \beta \in \mathbb{K}$, one must have $\alpha u + \beta v \in V$. It stands to reason that we might like to add or scale more than two vectors at a time! The following definition puts a name to a scaled sum of an arbitrary, finite collection of vectors.

Definition 1.2 (Linear Combination) Consider a vector space V over a field $\mathbb{K}$, and a finite collection $\{v_1, ..., v_k\} \subseteq V$ of vectors in V. A linear combination of the collection $\{v_1, ..., v_k\}$ is any vector of the form,

$$c_1 v_1 + ... + c_k v_k, \tag{1.7}$$

where $c_1, ..., c_k \in \mathbb{K}$ are any scalars in $\mathbb{K}$.

Remark 1.3 It's important to note that a linear combination of vectors is a scaled sum of a *finite* collection of vectors. We can take as many vectors as we want in a linear combination, as long as the number is not infinite.

There is a subtlety present in the definition of a linear combination that we've glossed over at first pass. The definition of a vector space states that a linear combination of *two* vectors will always belong to the vector space - how do we know that the linear combination of *any* finite linear combination of vectors will be in the vector space? Although this seems like a trivial detail, it's vital that we dot all of our i's and cross all of our t's before we begin using these concepts in earnest. Therefore, every time we make a new definition, we must ensure that the definition *well-posed* - that it is logically sound and doesn't lead to any contradictions in our theory. The following proposition confirms that the definition of a linear combination *is* in fact well-posed.

Proposition 1.1 (Vector Spaces are Closed Under Linear Combinations) *Consider a vector space V and a collection $\{v_1, ..., v_k\} \subseteq V$ of vectors in V. Any linear combination of $v_1, ..., v_k$ also belongs to V.*

Exercise 1.2 Prove Proposition 1.1. (Hint: use induction on k).

Oftentimes, we'll be interested in closely examining a special subset of a given vector space. If the subset of a given vector space *still* has the structure of a vector space, we call it a *subspace*. We make this idea explicit with the following definition.

Definition 1.3 (Subspace) Consider a vector space V over $\mathbb{K}$ with operations $+$ and $(\cdot)$. A subset $W \subseteq V$ is said to be a subspace of V if, under the operations $+$ and $(\cdot)$ of V, it is also a vector space over $\mathbb{K}$.

Taking a look at the definition of a subspace, it seems like it would be an awful lot of work to verify that a given subset of a vector space is in fact a subspace. Fortunately, there is a *much* easier way to verify a given set is a subspace than checking that all 10^n axioms of a vector space hold. Since a subspace is already a subset of a vector space, subspaces automatically inherit a number of the vector space properties. What remains to be checked is covered by the following proposition.

Proposition 1.2 *Consider a vector space V. A subset $W \subseteq V$ is a subspace of V if and only if, for all $\alpha, \beta \in \mathbb{K}$ and $u, v \in W$, $\alpha u + \beta v \in W$.*

In other words, a subset of a vector space is a subspace if and only if it is closed under linear combinations, with respect to the operations of the original vector space. All other properties of a vector space are inherited from the fact that a subspace is a subset of the vector space.

Exercise 1.3 Prove Proposition 1.2.

An important example of a subspace is the *span* of a collection of vectors, which we now define.

Definition 1.4 (Span) Consider a collection $\{v_1, ..., v_k\} \subseteq V$ of vectors. The span of $\{v_1, ..., v_k\}$, denoted $\text{span}\{v_1, ..., v_k\}$, is the set of all linear combinations of $v_1, ..., v_k$,

$$\text{span}\{v_1, ..., v_k\} = \{c_1 v_1 + ... + c_k v_k : c_i \in \mathbb{K}\} \subseteq V. \tag{1.8}$$

Exercise 1.4 Show that the span of a finite collection of vectors of a vector space V is a subspace of V.

Let's study some more basic properties of vector spaces and linear combinations. As you may recall, an important property that a collection of vectors can have is *linear independence.* Fundamentally, a collection of vectors is linearly independent if each vector "points in a new direction" compared to the other vectors in the collection. We formalize this intuitive notion of linear independence in the following definition.

Definition 1.5 (Linear Independence) Let V be a vector space over $\mathbb{K}$. Consider a collection $\{v_1, ..., v_k\} \subseteq V$ of vectors in V. The collection is said to be *linearly independent* if, for all $c_1, ..., c_k \in \mathbb{K}$ with $c_1, ..., c_k$ not all zero, $c_1 v_1 + ... + c_k v_k \neq 0$. If the collection is *not* linearly independent, it is said to be *linearly dependent.*

Let's verify that the formal definition of linear independence matches up with our intuitive notion of vectors "pointing in new directions." Suppose we're given a collection of linearly *dependent* vectors, $\{v_1, ..., v_k\}$. Then, by Definition 1.5, there exist constants $c_1, ..., c_k \in \mathbb{K}$, not all zero, for which

$$c_1 v_1 + c_2 v_2 + ... + c_k v_k = 0. \tag{1.9}$$

Without loss of generality, suppose that $c_1 \neq 0$ (the choice to focus on c_1 is arbitrary, as no c_i, v_i pair has any property making it more "special" than the others - we can therefore focus on c_1 *without loss of generality*). Then, by the above, we can write,

$$v_1 = -\frac{c_2}{c_1} v_2 - ... - \frac{c_k}{c_1} v_k. \tag{1.10}$$

Therefore, we conclude that v_1 "points in a direction" that is *already* covered by the remaining $k-1$ vectors. Thus, it is *not* true that in a linearly dependent set, every vector "points in a new direction." We conclude that the formal definition of linear independence is consistent with the intuitive notion we outlined above.

Armed with the definition of linear independence, we now have the ability to study a variety of more sophisticated constructions. First, we introduce the definition of a basis for a vector space.

Definition 1.6 (Basis) Consider a vector space V over $\mathbb{K}$. A basis $\mathcal{B}$ for V is a linearly independent collection, $\mathcal{B} = \{v_1, ..., v_k\} \subseteq V$, such that for all $v \in V$, there exist scalars $c_1, ..., c_k \in \mathbb{K}$ for which

$$v = c_1 v_1 + ... + c_k v_k. \tag{1.11}$$

Any vector $v_i \in \mathcal{B}$ belonging to the basis is called a basis vector.

Exercise 1.5 Let $\mathcal{B} = \{v_1, ..., v_k\}$ be a basis for a vector space V. Show that for any $v \in V$, the constants $c_1, ..., c_k \in \mathbb{K}$ for which $v = c_1 v_1 + ... + c_k v_k$ are uniquely determined by v.

Thus, a collection of vectors is a *basis* for a vector space if the collection is linearly independent and we can write any element of the vector space as a (unique) linear combination of the basis vectors. Eagle-eyed readers will note that here, we've defined a basis to be a *finite* collection of vectors! We'll address our reasoning for this shortly.

In order to understand why we've defined a basis in terms of a finite collection of vectors, we first need to define the *dimension* of a vector space. Recall that, given a set S with a finite number of elements, $|S|$ denotes the number of elements in S. With this in mind, we make the following definition.

Definition 1.7 (Dimension) Consider a vector space V over $\mathbb{K}$. Suppose $\mathcal{B} \subseteq V$ is a basis for V. The dimension of V, $\dim(V)$, is the number of elements in the basis, $\dim(V) = |\mathcal{B}|$.

In order for this definition to be well-defined, it's important that we check that *all bases* for a given vector space have the same dimension! The following result confirms that this is in fact the case.

Proposition 1.3 (Dimension is Well-Defined) *Consider a vector space V over $\mathbb{K}$. If $\mathcal{B}_1$ and $\mathcal{B}_2$ are two bases for V, then $|\mathcal{B}_1| = |\mathcal{B}_2|$.*

Let's discuss why Proposition 1.3 implies dimension is a well-posed quantity. Given any two bases for V, Proposition 1.3 states that the bases must contain the same number of elements. Since dimension is defined as the number of elements in a basis, Proposition 1.3 confirms that dimension is a property of a vector space, rather than a property of a basis. Thus, it makes sense to write $\dim(V)$, rather than $\dim(\mathcal{B})$.

Exercise 1.6 Prove Proposition 1.3.

Before we move on to the study of linear transformations, we have one more point to discuss regarding bases! In your first course in linear algebra, you likely worked exclusively with *finite-dimensional* vector spaces - vector spaces in which there exists a finite basis. What, then, does it mean for a vector space to be infinite-dimensional?

Definition 1.8 (Finite/Infinite-dimensional Vector Space) A vector space V is said to be finite-dimensional if it has a basis with a finite number of elements. If no such basis exists, V is said to be infinite-dimensional.

Defining bases for infinite-dimensional spaces is a somewhat more subtle problem; one that is beyond the scope of our brief review of linear algebra. We refer the interested reader to the references at the end of the chapter for a treatment of infinite-dimensional bases.

Now that we've studied vector spaces in reasonable detail, we can discuss linear transformations, which are special maps between vector spaces. From your first course in linear algebra, you might immediately think of a matrix when you think of a linear transformation. If the vector space you're working in is $\mathbb{R}^n$, this is certainly justifiable! However, in abstract vector spaces, matrices are *not* immediately associated with linear transformations, analogous to how vectors are not immediately associated with tuples of numbers. Consider the following, abstract definition.

Definition 1.9 (Linear Transformation) Consider two vector spaces V, W over $\mathbb{K}$. A linear transformation between V and W is a map $A : V \to W$ such that for all $\alpha, \beta \in \mathbb{K}$ and $u, v \in V$, $A(\alpha u + \beta v) = \alpha A(u) + \beta A(v)$.

Remark 1.4 Note that, instead of writing $A(u)$ for the action of a linear transformation $A : V \to W$ on a vector $u \in V$, it is convention to write Au. This convention *does not* extend to nonlinear maps.

Thus, an "abstract" linear transformation is simply a map between vector spaces that *respects* the linear structure of the vector spaces. The study of linear transformations between vector spaces is one that is surprisingly deep. We'll return to linear transformations after a brief digression into *analysis* on the real line and in vector spaces.

1.2 Crumbs of Real Analysis

In systems and control theory, bounds are used to estimate complex quantities in terms of simple ones. For instance, in Chapter 2, when we study stability, we'll look for *exponential bounds* on the trajectories of our system. To effectively study mathematical control theory, it's therefore important that we have the tools we need to bound sets of real numbers, and reason about when these bounds are *sharp*.

This is where real analysis comes in. Fundamentally, elementary real analysis is the centered around the study of *convergence* of sequences and the *shapes* of sets of real numbers. In this section, we'll focus on this second point, and develop the necessary concepts required to find sharp bounds on sets of real numbers.

First, we'll introduce some basic language and facts about subsets of the real line, $\mathbb{R}$, and then proceed to develop sharp bounds - called suprema and infima - on these sets. Finally, we'll briefly look at how these sharp bounds on sets interact with functions. As a first step towards achieving these goals, we define what it means for a set to be *bounded*.

Definition 1.10 (Bounded Above/Below) Consider a subset $A \subseteq \mathbb{R}$. A is said to be:

1. Bounded above: if there exists an $U \in \mathbb{R}$ such that $x \leq M$, for all $x \in A$. In this case, such a U is said to be an upper bound on A.
2. Bounded below: if there exists an $L \in \mathbb{R}$ such that $L \leq x$, for all $x \in A$. In this case, such an L is said to be a lower bound on A.
3. Bounded: if there exists an $M \in \mathbb{R}$ such that $|x| \leq M$, for all $x \in A$.

The definitions presented above simply tell us whether or not a set is bounded. Notably, we do *not* specify how close or far our upper or lower bounds are from the set. For instance, if one takes the interval $[0, 1) \subseteq \mathbb{R}$, both 1 and 100 are equally valid upper bounds. However, the upper bound of 1 *clearly* provides us with more information about the set than the upper bound of 100. Now, we seek a bound on a set that provides us with the *most possible information* about a set - i.e. a bound that is as tight as possible. Consider the following definitions.

Definition 1.11 (Supremum) Consider a set $A \subseteq \mathbb{R}$. The supremum of A, denoted $\sup A$, is the least upper bound of A. That is, if $U \in \mathbb{R}$ is any upper bound on A, $\sup A \leq U$.

Just as we define the least upper bound, we also define the greatest lower bound.

Definition 1.12 (Infimum) Consider a set $A \subseteq \mathbb{R}$. The infimum of A, denoted $\inf A$, is the greatest lower bound of A. That is, if L is any lower bound on A, then $L \leq \inf A$.

The supremum and infimum are defined to be the *tightest possible* upper and lower bounds on a given subset of $\mathbb{R}$. Let's look at a couple of quick examples.

Example 1.4 Consider the set $A = [a, b) \subseteq \mathbb{R}$, where $a < b$. Here, $\sup A = b$, as b is the smallest possible upper bound on A. Likewise, $\inf A = a$, since a is the greatest possible lower bound on A.

This example highlights an important feature of suprema and infima - it is *not* necessarily the case that $\sup A \in A$ or $\inf A \in A$. In the example of $[a, b)$, the infimum belonged to the set, while the supremum did not. Let's take a second look at the definitions of the supremum and infimum to see why this is. When looking at the definitions of suprema and infima, one makes a natural comparison to the maximum and minimum of a set. Let's make a formal definition of a maximum and minimum to clear up the difference between a maximum/supremum and minimum/infimum.

Definition 1.13 (Maximum/Minimum) Consider a subset $A \subseteq \mathbb{R}$. A point $a \in A$ is said to be the maximum element of A if $x \leq a$ for all $x \in A$. Likewise, a point $b \in A$ is said to be the minimum element of A if $b \leq x$ for all $x \in A$.

Thus, we observe that unlike the supremum and infimum, the maximum or minimum of a set *must belong* to the set. As a consequence of this, given an arbitrary subset $A \subseteq \mathbb{R}$, one is *not* guaranteed to have a maximum or minimum value, even if it is bounded above and below! Consider, for example, the set $(a, b) \subseteq \mathbb{R}$. Here, the set is bounded above and below *but* has no maximum or minimum value, since a and b are not included. However, both the supremum and infimum exist; $\sup(a, b) = b$ and $\inf(a, b) = a$.

In the event where they exist, how do the maximum and minimum of a set relate to the supremum and infimum? The following result provides an answer.

Proposition 1.4 *Consider a set $A \subseteq \mathbb{R}$. If A has a maximum element, then* $\max A = \sup A$. *Likewise, if A has a minimum element, then* $\min A = \inf A$.

Thus, as our intuition might confirm, the maximum and minimum coincide with the supremum and infimum *when they exist.*

Exercise 1.7 Prove Proposition 1.4.

The question of existence of maxima and minima naturally begs the question - do the supremum and infimum of a set always exist? The following fact answers this question.

Fact (Axiom of the Supremum) A nonempty, bounded above set $A \subseteq \mathbb{R}$ has a finite supremum, $\sup A < \infty$. □

Notably, we state this result as a *fact*, not as a proposition! Why is this? As it happens, the axiom of the supremum is baked into the formal definition of the real numbers, $\mathbb{R}$.[1] As such, it is not something that we formally need to prove. In the case where A is *not* bounded above, we take $\sup A = \infty$ by convention. In the case where $A = \emptyset$, we take $\sup A = -\infty$ by convention.

This answers the question of existence of the supremum. What about existence of the infimum? In order to answer this question, we state a handy proposition which lets us translate results about the supremum to results about the infimum.

[1] This is a component of one of a number of equivalent, formal definitions for $\mathbb{R}$. The interested reader is encouraged to consult the references provided at the end of the chapter to learn more about the construction of the reals.

Proposition 1.5 *For a set $A \subseteq \mathbb{R}$, $\inf A = -\sup(-A)$, where $-A := \{-x : x \in A\}$.*

This result enables us to directly translate results about the supremum to results about the infimum, simply by flipping the sign of elements in the set. Generally, we'll prove results for the supremum and translate them to the infimum using Proposition 1.5.

Exercise 1.8 Prove Proposition 1.5.

By Proposition 1.5 and the Axiom of the Supremum, the following result is immediate.

Proposition 1.6 *Any nonempty, bounded below set $A \subseteq \mathbb{R}$ has a finite infimum.*

Using our conventions for the supremum along with Proposition 1.5, we have that $\inf A = -\infty$ when A is not bounded below and that $\inf A = \infty$ when $A = \emptyset$.

Thus far, we've only discussed the suprema and infima of generic subsets of $\mathbb{R}$. Now, we'll consider how suprema and infima interact with real-valued functions. Although this might initially seem like a step up from working with suprema and infima of sets, all we need to do to treat suprema and infima of functions is introduce a little bit of notation. Consider an arbitrary set A (not even necessarily a subset of $\mathbb{R}$), and a function $f : A \to \mathbb{R}$. We define,

$$\sup_{x \in A} f(x) := \sup\{f(x) : x \in A\} = \sup f(A) \tag{1.12}$$

$$\inf_{x \in A} f(x) := \inf\{f(x) : x \in A\} = \inf f(A). \tag{1.13}$$

Thus, taking the supremum and infimum of real-valued functions is really the same thing as taking the supremum and infimum of sets - we simply take the supremum and infimum of the *images* of sets, which are nothing more than standard subsets of $\mathbb{R}$. Now that we've discussed the foundational aspects of suprema and infima, we state a number of their basic properties.

Proposition 1.7 (Properties of Suprema) *Consider sets $A, B \subseteq \mathbb{R}$. The suprema of A and B satisfy the following properties.*

1. *Subset Inequality: If $B \subseteq A$, then $\sup B \leq \sup A$.*
2. *Sum of Sets Equality: If A, B are nonempty, then $\sup(A + B) = \sup A + \sup B$, where $A + B$ is defined, $A + B = \{a + b : a \in A, b \in B\}$.*
3. *Sum of Functions Inequality: For D an arbitrary set and $f, g : D \to \mathbb{R}$ functions,*

$$\sup_{x \in D}(f(x) + g(x)) \leq \sup_{x \in D} f(x) + \sup_{x \in D} g(x). \tag{1.14}$$

It's extremely important that we distinguish between properties (2) and (3) listed above. Although it might initially seem that the same rule would apply for functions and sets, this is not the case! The intuitive reasoning for this is as follows. For functions, taking the sum $\sup_{x \in D} f(x) + g(x)$ means taking the supremum of the sum where f and g are evaluated at the *same* point x. However, taking the sum $\sup_{x \in D} f(x) + \sup_{x \in D} g(x)$ means that f and g can take in different values! This yields an inequality, as both f and g are free to individually take on their suprema. Note that this reasoning takes a little bit of sharpening up to yield a formal proof - in particular, one can formally prove (3) by applying (1) and (2). We leave the details of this task as an exercise.

Exercise 1.9 Apply Proposition 1.5 to reformulate Proposition 1.7 in terms of infima.

Exercise 1.10 Supply a formal proof of item (3) of Proposition 1.7 using items (1) and (2).

1.3 Normed Vector Spaces

Now that we've sketched out some basic real analysis, we return to the domain of linear algebra and study basic analysis in vector spaces. First, we define a norm on a vector space.

Definition 1.14 (Norm) Consider a vector space V over $\mathbb{K}$. A norm is a map $\|\cdot\| : V \to \mathbb{R}_{\geq 0}$ satisfying the following conditions:

1. Positive Definite: $\|u\| \geq 0$ for all $u \in V$, and $\|u\| = 0$ if and only if $u = 0$.
2. Positive Homogeneity: For all $u \in V$ and $c \in \mathbb{K}$, $\|cu\| = |c| \, \|u\|$.
3. Triangle Inequality: For all $u, v \in V$, $\|u + v\| \leq \|u\| + \|v\|$.

Remark 1.5 Here, we use $|\cdot|$ to denote the magnitude of a scalar. For $\mathbb{K} = \mathbb{R}$, this is equal to the absolute value, and for $\mathbb{K} = \mathbb{C}$, this is equal to the complex magnitude.

Since $\|\cdot\| : V \to \mathbb{R}_{\geq 0}$ maps to the positive reals (which *do not* include ∞), it is a requirement that the norm of any given vector is finite! If $\|v\|$ is not finite for some $v \in V$, then $\|\cdot\|$ is *not* a valid norm on V. Let's consider a few common examples of norms on $\mathbb{R}^n$. In each of the following examples, convince yourself that each norm is finite for all vectors in the vector space on which they are defined.

Example 1.5 (ℓ^2 Norm) The ℓ^2 norm on $\mathbb{R}^n$, alternatively called the 2-norm or Euclidean norm, is defined

$$\|x\|_2 = \sqrt{x_1^2 + ... + x_n^2} = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{1.15}$$

Example 1.6 (ℓ^1 Norm) The ℓ^1 norm on $\mathbb{R}^n$, alternatively called the 1-norm, is defined

$$\|x\|_1 = |x_1| + ... + |x_n| = \sum_{i=1}^{n} |x_i|. \tag{1.16}$$

Example 1.7 (ℓ^∞ Norm) The ℓ^∞ norm on $\mathbb{R}^n$, alternatively called the ∞-norm, is defined

$$\|x\|_\infty = \max_{i \in 1, \ldots, n} |x_i|. \tag{1.17}$$

Exercise 1.11 Show that the $\ell^1, \ell^2, \ell^\infty$ norms proposed above are indeed norms on $\mathbb{R}^n$.

With the definition of a norm in hand, we're ready to define a *normed vector space*. Scary as this might sound, the definition of a normed vector space is actually quite innocent.

Definition 1.15 (Normed Vector Space) A normed vector space is a pair $(V, \|\cdot\|)$ of a vector space V and a norm $\|\cdot\|$ on V.

That was easy! Let's consider a couple of basic examples of normed vector spaces.

Example 1.8 $(\mathbb{R}, |\cdot|)$, the real line equipped with the absolute value, is perhaps the simplest example of a normed vector space. Similarly, $(\mathbb{C}, |\cdot|)$, the complex numbers equipped with the complex magnitude, also forms a normed vector space. Note that if we refer to $\mathbb{R}$ or $\mathbb{C}$ as normed vector spaces we will *always* assume the norms in question are the absolute value and complex magnitude, respectively, unless directed otherwise.

Example 1.9 $(\mathbb{R}^n, \|\cdot\|_1)$, $(\mathbb{R}^n, \|\cdot\|_2)$, and $(\mathbb{R}^n, \|\cdot\|_\infty)$ are all normed vector spaces.

Example 1.10 $(\mathbb{R}^{n\times n}, \|\cdot\|_2)$, the vector space of $n \times n$ matrices with real values, along with the matrix 2-norm, $\|A\|_2 = \sigma_{\max}(A)$ is a normed vector space.

Example 1.11 $(\mathbb{R}^{n\times n}, \|\cdot\|_F)$, the vector space of $n \times n$ matrices with real values, along with the Frobenius norm, $\|A\|_F = \sqrt{\operatorname{tr}(A^\top A)}$, is a normed vector space.

These examples are all fairly simple in nature - we take a vector space like $\mathbb{R}^n$ or $\mathbb{R}^{n\times n}$, which we understand well, and place a norm on it that is easy to compute in terms of the entries of the vector. Shortly, we'll discuss some more complex, infinite-dimensional examples. Before we can do this, however, we first need to learn some more analysis.

When we first took a crack at real analysis, we studied different subsets of the real line. We'll begin our study of analysis in normed vector spaces in the same spirit. We start by introducing a special subset of a normed vector space, called an *ϵ-ball.*

Definition 1.16 (Epsilon Ball) Consider a normed vector space $(V, \|\cdot\|)$. Let $\epsilon > 0$ be a fixed real number and $v \in V$ a vector. The ϵ-ball centered at v is the set,

$$B_\epsilon(v) := \{u \in V : \|u - v\| < \epsilon\}. \tag{1.18}$$

Remark 1.6 An ϵ-ball is also commonly referred to as an *ϵ-neighborhood.*

Let's make a few comments on this definition. First, we note that an epsilon ball *doesn't* include points that exactly a distance ϵ away from v; rather it only includes points that are strictly *less* than ϵ away from v. Thus, an ϵ ball has a "fuzzy" boundary. Secondly, we note that the actual shape of an ϵ-ball depends on the choice of norm! For instance, an ϵ-ball in $(\mathbb{R}^2, \|\cdot\|_2)$ will look like an actual ball, while an ϵ ball in $(\mathbb{R}^2, \|\cdot\|_\infty)$ will look like a square.

Exercise 1.12 Sketch the epsilon ball $B_1(0)$ on a pair of coordinate axes in the spaces $(\mathbb{R}^2, \|\cdot\|_1)$, $(\mathbb{R}^2, \|\cdot\|_2)$, and $(\mathbb{R}^2, \|\cdot\|_\infty)$. *Hint: the boundaries of these sets should be "fuzzy."*

Above, we mentioned that an ϵ ball has a "fuzzy" boundary - i.e. $B_\epsilon(v)$ does not have a sharp boundary where the set ends. We know that ϵ-balls are not the only sets with "fuzzy" boundaries we can draw - any set without a sharp boundary fits into this category. How, then, can we specify a general class of sets in a normed vector space with a "fuzzy" boundary? Consider the following definition.

Definition 1.17 (Open Set) Let $(V, \|\cdot\|)$ be a normed vector space. A subset $A \subseteq V$ is said to be an open set if, for all $v \in V$, there exists an $\epsilon > 0$ such that $B_\epsilon(v) \subseteq A$.

Remark 1.7 Note that a few expressions are commonly used to declare a set $A \subseteq V$ is open. The expressions "A is an open set," "A is open" (if the space V is clear), or "A is open in V" (if one wishes to emphasize the vector space), are all equivalently used to declare that a set A is open.

Thus, we declare a set $A \subseteq V$ to be *open* if, around each point in A, we can squeeze in an ϵ-ball that is still contained in the set. Open sets are the natural way of making precise the idea of a set with a "fuzzy" boundary. Let's run through a couple of examples of open sets.

Example 1.12 (Examples of Open Sets)

1. Any open interval, $(a, b),\ a < b$, is an open set in $(\mathbb{R}, |\cdot|)$.

2. Any union of open intervals, $(a, b) \cup (c, d),\ a < b,\ c < d$, is an open set in $(\mathbb{R}, |\cdot|)$.
3. In a normed vector space $(V, \|\cdot\|)$ any open ball $B_\epsilon(v)$ is open.

Exercise 1.13 Confirm that the examples above are indeed open sets. Sketch out each example and confirm that the sets have "fuzzy boundaries."

Now that we've seen some basic examples of open sets, let's outline some basic properties of open sets.

Proposition 1.8 (Properties of Open Sets) *Let $(V, \|\cdot\|)$ be a normed vector space. The open subsets of V satisfy the following properties:*

1. *Nothing & everything: $\emptyset$ and V are open sets.*
2. *Stability under unions: For $\{U_\alpha\}_{\alpha\in\Lambda}$ an arbitrary collection of open sets, the union $\cup_{\alpha\in\Lambda}U_\alpha$ is an open set.*
3. *Stability under finite intersections: For $\{U_i\}_{i=1}^k$ a finite collection of open sets, the intersection $\cap_{i=1}^k U_i$ is an open set.*

Remark 1.8 Just like we write $\{U_i\}_{i=1}^k$ to refer to the collection $\{U_1, ..., U_k\}$ of k sets, we use the notation $\{U_\alpha\}_{\alpha\in\Lambda}$ to refer to an arbitrary collection of sets, indexed by an arbitrary set Λ. Here, α is the index of an individual set in the collection (analogous to i), and Λ is the set of all indices of the collection (analogous to $\{1, ..., k\}$). For instance, if one has a collection of sets corresponding to real numbers, one might write $\{U_\alpha\}_{\alpha\in\mathbb{R}}$. When the index set is clear or unimportant, we will simply write $\{U_\alpha\}$ as shorthand to refer to an arbitrary collection of sets.

Let's review what each condition of the proposition says. The first condition, *nothing & everything*[2], states that the empty set (nothing) and the entire vector space V (everything) are both open sets. The empty set trivially satisfies Definition 1.17, since it has no points to check for, and V satisfies Definition 1.17 since any ϵ-ball is automatically contained in V.

The next condition, stability under unions, states that an *arbitrary* (potentially uncountably infinite) collection of open sets has a union that is also open. Why is this? If we pick a point $v \in \cup_\alpha U_\alpha$, the definition of a union tells us there exists an α for which $v \in U_\alpha$. Since U_α is open, there exists a ball $B_\epsilon(v) \subseteq U_\alpha \subseteq \cup_\alpha U_\alpha$. Thus, the union is open.

The final condition, stability under finite intersections, states that if we have a *finite* collection of open sets, their intersection must be open. The formal argument for this case takes a little more thought than stability under unions - we leave its proof as an exercise (with a hint) below. Why can't we take infinite intersections? Consider the following counterexample. Define a collection $\{B_{1/n}(0)\}_{n\in\mathbb{N}}$, of ϵ-balls centered at the origin with shrinking radius $1/n$. With a little work, one may show that,

$$\bigcap_{n=1}^{\infty} B_{1/n}(0) = \{0\}, \tag{1.19}$$

since all of the sets in the collection shrink down towards zero as $n \to \infty$. Since $\{0\}$ isn't an open set (we can't fit an epsilon ball of positive radius around 0 into the set $\{0\}$), this yields an example of an infinite collection of open sets whose intersection is *not* open. This results in condition (3) holding only for finite intersections.

[2] The nice terminology "nothing and everything" is due to Joel Tropp.

Exercise 1.14 Prove item (2) of Proposition 1.8. Hint: to pick the radius of an ϵ ball that fits in the intersection, try experimenting with the *minimum* of a set of epsilons.

Now that we've defined an open set, a natural question to ask is - what, if anything, is a *closed* set? If an open set is a set with a fuzzy boundary, perhaps a closed set should be a set with a sharp boundary. Although this intuitive definition covers a variety of closed sets, it isn't quite expressive enough to capture everything we need (for instance, are $\emptyset$ and V closed sets?). Consider the following, abstract definition.

Definition 1.18 (Closed Set) Let $(V, \|\cdot\|)$ be a normed vector space. A subset $A \subseteq V$ is said to be a closed set if its complement, $A^c = V \setminus A$, is an open set.

Let's think for a moment about why this definition might match up with our intuition regarding closed sets having "sharp" boundaries. If a set has a fuzzy boundary, then it stands to reason that its complement should have a sharp boundary. Likewise, if a set has a sharp boundary, then its complement should have a fuzzy boundary. Thus, a set with a sharp boundary should be closed.

Exercise 1.15 Draw some pictures to reconcile the "sharp boundary" intuition behind closed sets with the formal definition.

Let's explore some basic properties of closed sets.

Proposition 1.9 (Examples & Properties of Closed Sets) *Let $(V, \|\cdot\|)$ be a normed vector space. The closed subsets of V satisfy the following:*

1. *Nothing & Everything: $\emptyset$ and V are closed sets.*
2. *Stability under intersections: For $\{C_\alpha\}_{\alpha\in\Lambda}$ an arbitrary collection of closed sets, the intersection $\cap_{\alpha\in\Lambda} C_\alpha$ is a closed set.*
3. *Stability under finite unions: For $\{C_i\}_{i=1}^k$ a finite collection of closed sets, the union $\cup_{i=1}^k C_i$ is a closed set.*

Thus, we observe that closed sets seem to satisfy the exact *opposite* properties of open sets! The reason for this is precisely that closed sets are defined as complements of open sets - the complement *flips* the properties of open sets to properties of closed sets.

Exercise 1.16 Use DeMorgan's laws to prove Proposition 1.9 directly from Proposition 1.8.

Exercise 1.17 Produce an infinite collection of closed sets whose union is not closed.

Closed sets have a number of convenient properties that makes them easy to work with when considering matters such as continuity and convergence. As such, given an arbitrary subset A of a normed vector space V, we often wants to find the "smallest" closed set containing A. This way, we can preserve the basic structure of A while gaining the extra properties of a closed set. Consider the following definition, which defines the "smallest" closed set containing any given set.

Definition 1.19 (Closure) Let $(V, \|\cdot\|)$ be a normed vector space and $A \subseteq V$ an arbitrary subset. The closure of A, denoted $\overline{A}$, is the smallest closed set containing A,

$$\overline{A} = \bigcap_{\alpha\in\Lambda} C_\alpha, \text{ where } \{C_\alpha\}_{\alpha\in\Lambda} = \{C_\alpha \subseteq V : A \subseteq C_\alpha \text{ and } C_\alpha \text{ is closed}\}. \tag{1.20}$$

As the closure of A is defined as the *intersection* of all closed sets containing A, one can think of the closure of A as "shrink wrapping" the set A with closed sets. The following two exercises provide some quick sanity checks regarding the closure.

Exercise 1.18 Verify that the closure of any given set is in fact closed.

Exercise 1.19 Show that a set $A \subseteq V$ is closed if and only if $A = \overline{A}$.

The following is a nice consequence of these two exercises.

Example 1.13 Consider a normed vector space $(V, \|\cdot\|)$. The closure of any open ball $B_\epsilon(x)$ in V is the closed ball,

$$\overline{B}_\epsilon(x) = \{y \in V : \|x - y\| \leq \epsilon\}. \tag{1.21}$$

Thus, the closure makes the ordinarily "fuzzy" boundary of an open ball a "sharp" boundary. Try sketching out the intersections of some closed sets containing the ball $B_\epsilon(x)$ to convince yourself that this also follows from the definition of the closure.

Just as we can define the smallest closed set containing a given set, we can define the *largest open set* contained within a given set.

Definition 1.20 (Interior) Let $(V, \|\cdot\|)$ be a normed vector space and $A \subseteq V$ an arbitrary subset. The interior of A, denoted A°, is the largest open set contained in A,

$$A^\circ = \bigcup_{\alpha \in \mathit{\Lambda}} O_\alpha, \text{ where } \{O_\alpha\}_{\alpha \in \mathit{\Lambda}} = \{O_\alpha \subseteq V : O_\alpha \subseteq A \text{ and } O_\alpha \text{ is open}\}. \tag{1.22}$$

Instead of "shrink wrapping" a set with a collection of closed sets, as we did in the case of the closure, we can think of the interior as inflating a collection of open sets inside the given set, until we fill up the entire inside space with open sets. What remains after this operation is the largest open set contained inside the given set.

Exercise 1.20 Verify that the interior of any given set is in fact open.

Exercise 1.21 Show that a set $A \subseteq V$ is open if and only if $A = A^\circ$.

Example 1.14 Consider a normed vector space $(V, \|\cdot\|)$. The interior of any closed ball $\overline{B}_\epsilon(x)$ in V is the open ball, $B_\epsilon(x)$. Thus, the interior makes the ordinarily "sharp" boundary of a closed ball a "fuzzy" boundary.

So far, we've only examined normed vector spaces $(V, \|\cdot\|)$ with a fixed choice of norm on V. Now, we examine what happens when we change the norm on a given vector space. What effect does the choice of norm $\|\cdot\|$ on V have on the properties of the normed vector space $(V, \|\cdot\|)$? As we saw in our definition of an open set above, the choice of a norm on a vector space *entirely* determines which sets are open and closed. Thus, to determine the effect of a norm on the vector space, it's logical to ask - when do two norms on a single vector space determine the same open sets?

As a first step towards answering this question, we define what it means for two norms on the same vector space to be *equivalent*.

Definition 1.21 (Equivalent Norms) Consider a vector space V. Two norms, $\|\cdot\|_a$ and $\|\cdot\|_b$, on V are said to be equivalent if there exist constants $k_1, k_2 > 0$ for which

$$k_1 \|v\|_a \leq \|v\|_b \leq k_2 \|v\|_a, \; \forall v \in V. \tag{1.23}$$

Thus, two norms on a vector space are said to be equivalent if one norm can be "sandwiched" between some positive multiples of the other. We now show that the equivalence of two norms *entirely* determines whether the open sets of the normed vector spaces determined by the norms will be the same.

Proposition 1.10 (Equivalent Norms Determine the Same Open Sets) *Consider a vector space V and norms $\|\cdot\|_a$ and $\|\cdot\|_b$ on V. The open sets of $(V, \|\cdot\|_a)$ and $(V, \|\cdot\|_b)$ are the same if and only if $\|\cdot\|_a$ and $\|\cdot\|_b$ are equivalent norms.*

Proof First, suppose $\|\cdot\|_a$ and $\|\cdot\|_b$ are equivalent. Consider an arbitrary open ball $B^a_{\epsilon_0}(x)$, with respect to the norm $\|\cdot\|_a$. By the definition of norm equivalence, it follows that there exists an open ball $B^b_{\epsilon_1}(x)$ with respect to the norm $\|\cdot\|_b$, satisfying $B^b_{\epsilon_1}(x) \subseteq B^a_{\epsilon_0}(x)$.

Now, let U_a be an arbitrary open set of $(V, \|\cdot\|_a)$. We aim to show that U_a is also open in $(V, \|\cdot\|_b)$. Let $x \in U_a$ be arbitrary. Since U_a is open in x, there exists a ball $B^a_\epsilon(x) \subseteq U_a$. But, by the reasoning above, there exists a ball $B^b_\epsilon(x) \subseteq B^a_\epsilon(x) \subseteq U_a$. So, around every point in U_a, we can squeeze in a ball defined with respect to $\|\cdot\|_b$. We conclude that U_a is open in $(V, \|\cdot\|_b)$. To show that any open set U_b in $(V, \|\cdot\|_b)$ is also open in $(V, \|\cdot\|_a)$, one follows the same reasoning.

Now, we prove the opposite direction. For this direction, all we have to work with are open sets. We'll have to be a little bit clever, and pick our open sets such that they directly give us the constants we need for norm equivalence. Suppose the open sets of $(V, \|\cdot\|_a)$ and $(V, \|\cdot\|_b)$ are the same - i.e. a set $U \subseteq V$ is open in $(V, \|\cdot\|_a)$ if and only if it is open in $(V, \|\cdot\|_b)$. Consider the open ball of radius 1 around the origin in $\|\cdot\|_b$, denoted $B^b_1(0)$. Since $B^b_1(0)$ is open in $\|\cdot\|_b$, it must also be open in $\|\cdot\|_a$. Therefore, there exists a ball $B^a_{\epsilon_1}(0)$ in $\|\cdot\|_a$ satisfying $B^a_{\epsilon_1}(0) \subseteq B^b_1(0)$. There also exists a ball $B^a_{\epsilon_2}(0)$ in $\|\cdot\|_a$ satisfying $B^b_1(0) \subseteq B^a_{\epsilon_2}(0)$. The inclusion of balls,

$$B^a_{\epsilon_1}(0) \subseteq B^b_1(0) \subseteq B^a_{\epsilon_2}(0), \tag{1.24}$$

implies that for any vector $v \in V$ satisfying $\|v\|_b = 1$,

$$\epsilon_1 \|v\|_a \leq 1 \leq \epsilon_2 \|v\|_a . \tag{1.25}$$

Now, consider an arbitrary, nonzero vector $v \in V$. By the positive homogeneity property of norms, it follows that,

$$\epsilon_1 \left\| \frac{v}{\|v\|_b} \right\|_a \leq \frac{\|v\|_b}{\|v\|_b} \leq \epsilon_2 \left\| \frac{v}{\|v\|_b} \right\|_a \tag{1.26}$$

$$\frac{\epsilon_1}{\|v\|_b} \|v\|_a \leq \frac{\|v\|_b}{\|v\|_b} \leq \frac{\epsilon_2}{\|v\|_b} \|v\|_a \tag{1.27}$$

$$\epsilon_1 \|v\|_a \leq \|v\|_b \leq \epsilon_2 \|v\|_a . \tag{1.28}$$

For the remaining case of $v = 0$, the final inequality above holds trivially. We conclude that $\|\cdot\|_a$ and $\|\cdot\|_b$ are in fact equivalent. □

The next result - the proof of which we leave to the problems at the end of the section - is one of the most fundamental results in analysis in normed vector spaces. It tells us that, in any finite-dimensional vector space, *all* norms are equivalent. Because of this fact, the choice of norm in a finite-dimensional vector space often isn't critical - any two norms on a finite-dimensional vector space will produce normed vector spaces with similar analytical properties.

Theorem 1.1 (Norm Equivalence in Finite Dimensions) *Consider a finite-dimensional vector space V. Any two norms $\|\cdot\|_a$ and $\|\cdot\|_b$ on V are equivalent.*

Proof See [4] for the details. □

Now that we've discussed the structure of some important subsets of normed vector spaces, we turn our attention to the *continuity* of maps between normed vector spaces. Consider the following definition.

Definition 1.22 (Continuity) Consider two normed vector spaces, $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$. A mapping $f : V \to W$ is continuous at a point $x \in V$ if, for all $\epsilon > 0$, there exists a $\delta > 0$ (possibly a function of ϵ and x) for which

$$\|x - y\|_V < \delta \implies \|f(x) - f(y)\|_W < \epsilon. \tag{1.29}$$

If $f : V \to W$ is continuous at all x in a subset $U \subseteq V$, f is said to be continuous on U. Likewise, if f is continuous at all $x \in V$, it is simply said to be continuous.

Let's briefly check our understanding of this definition. Definition 1.22 states that, if a mapping $f : V \to W$ is continuous, $f(x)$ can change by a small amount if x changes by a small amount. It's important to note - the norm on $x - y$ is the norm $\|\cdot\|_V$ (since $x, y \in V$), while the norm on $f(x) - f(y)$ is the norm $\|\cdot\|_W$ (since $f(x), f(y) \in W$). One must take special care to ensure the correct norms are being used on $x - y$ and $f(x) - f(y)$ - these norms will *not* in general be the same!

Proposition 1.11 (Properties of Continuous Functions) *Let $(U, \|\cdot\|_U)$, $(V, \|\cdot\|_V)$, and $(W, \|\cdot\|_W)$ be normed vector spaces over $\mathbb{K}$ and $f : V \to W$, $g : V \to W$, and $h : W \to U$ be continuous mappings.*

1. *Algebraic Combinations: For any continuous functions $\alpha, \beta : V \to \mathbb{K}$, the function $p : V \to W$, defined $p(x) = \alpha(x) \cdot f(x) + \beta(x) \cdot g(x)$, is also continuous.*
2. *Composition: The composition $h \circ f : V \to U$ is continuous.*

Remark 1.9 In item (1) of the proposition above, we use continuous functions $\alpha, \beta : V \to \mathbb{K}$. Here, we treat $\mathbb{K}$ as the normed vector space $(\mathbb{K}, |\,\|\cdot\|\,|)$, where $|\cdot|$ represents either absolute value (for $\mathbb{K} = \mathbb{R}$) or complex magnitude (for $\mathbb{K} = \mathbb{C}$). It's necessary to treat $\mathbb{K}$ as a normed vector space in order to apply Definition 1.22!

When studying a function $f : V \to \mathbb{R}$ from a normed vector space V to the real line $\mathbb{R}$, it's useful to have some criteria to determine whether or not the function attains a maximum or minimum value on a given set. A special class of sets - termed *compact sets* - enable exactly this ability, among many others. In order to provide a sufficiently abstract definition of a compact set, we first require the definition of an open cover.

Definition 1.23 (Cover/Open Cover) Consider a normed vector space $(V, \|\cdot\|)$ and a subset $A \subseteq V$. A collection $\{U_\alpha\}_{\alpha \in \Lambda}$ of subsets of V is said to be a *cover* of A if

$$A \subseteq \bigcup_{\alpha \in \Lambda} U_\alpha. \tag{1.30}$$

If each U_α is open, the collection $\{U_\alpha\}_{\alpha \in \Lambda}$ is said to be an *open cover* of A. A cover is said to be *finite* if it contains a finite number of sets.

Remark 1.10 Sometimes, one will encounter the phrase, "$\{U_\alpha\}_{\alpha\in\Lambda}$ *covers* A." This is simply another way of saying that $\{U_\alpha\}_{\alpha\in\Lambda}$ is a cover of A.

This definition tells us that a *cover* of a given set is a collection of sets that, when pasted together, contain the given set.

Definition 1.24 (Subcover) Consider a normed vector space $(V, \|\cdot\|)$, a subset $A \subseteq V$, and a cover $\{U_\alpha\}_{\alpha\in\Lambda}$ of A. A subcollection $\{V_\beta\}_{\beta\in B} \subseteq \{U_\alpha\}_{\alpha\in\Lambda}$ is said to be a subcover of $\{U_\alpha\}_{\alpha\in A}$ if $\{V_\beta\}_{\beta\in B}$ is still a cover of A. A subcover is said to be *finite* if it contains a finite number of sets.

Remark 1.11 It's extremely important to note that $\{V_\beta\}_{\beta\in B} \subseteq \{U_\alpha\}_{\alpha\in\Lambda}$ *does not* mean that the sets V_β are subsets of the sets U_α. Here, the subset relation is a relation on the *collections* $\{V_\beta\}$ and $\{U_\alpha\}$ - we pick out a few of the elements of the collection $\{U_\alpha\}$ to form $\{V_\beta\}$. If the indices of the sets are not changed when one picks these elements out, one will have $B \subseteq \Lambda$.

Remark 1.12 Notice how the use of the prefix *sub* mirrors that of a subspace. Generally a sub-"object" is a subset of an "object" that retains the object's key properties. This is true of a subspace, wherein we have a subset of vector space that remains a vector space, and of a subcover, wherein we have a subset of a cover that remains a cover.

Thus, a subcover of a cover of a given set takes picks out a few sets from the cover that, when pasted together, still cover the given set. Armed with these definitions, we're ready to state an abstract definition of a compact set.

Definition 1.25 (Compact Set) Let $(V, \|\cdot\|)$ be a normed vector space. A subset $K \subseteq V$ is said to be a compact set if every open cover of K has a finite subcover.

Remark 1.13 To say that a set $K \subseteq V$ is a compact set, one will often say "K is compact," or "K is compact in V." This is just like how, to declare a set O is an open set, we said "O is open," or "O is open in V."

At first glance, this definition seems entirely mystifying. In order to pull back the curtain on compactness, we'll study a few basic results of compact sets. The following result tells us a few properties that compact sets must satisfy in a normed vector space. Note that the proofs of the next few compactness results are generally out of scope of our treatment of the material - one may consult the references at the end of the chapter for their proofs.

Proposition 1.12 (Compact Sets are Closed & Bounded) *Let $(V, \|\cdot\|)$ be a normed vector space and $K \subseteq V$ be a compact subset of V. Then, K must satisfy the following two properties:*

1. *Closed: K is closed in V.*
2. *Bounded: K is bounded in V - there exists an $M \geq 0$ such that $\|x\| \leq M$ for all $x \in M$.*

It's natural to wonder whether the converse of the result above is true - is every closed & bounded set compact? In *finite-dimensional* normed vector spaces, this turns out to be true, but in infinite-dimensional spaces, this is generally false. We postpone a more complete characterization of the infinite-dimensional case to our later discussion of Banach spaces. For now, we content ourselves with the (very nice) finite-dimensional result.

Proposition 1.13 (Compact Sets in Finite-Dimensional Spaces) *Consider a normed vector space $(V, \|\cdot\|)$. If V is finite-dimensional, then a subset $K \subseteq V$ is compact if and only if it is closed & bounded.*

This proposition produces a quick and easy way to verify a given set in a finite-dimensional vector space is compact. In the following two examples, we illustrate how easy it is to determine some simple subsets of a finite-dimensional space are compact.

Example 1.15 In $\mathbb{R}$, any closed interval $[a, b]$, $a \leq b$ is compact, since any closed interval is closed and bounded.

Example 1.16 In $\mathbb{R}^n$, any closed box $[a, b]^n = \{x \in \mathbb{R}^n : a \leq x_i \leq b\}$, $a \leq b$, is compact, since closed boxes are closed and bounded.

Example 1.17 In a finite-dimensional normed vector space $(V, \|\cdot\|)$, any closed ball $\overline{B}_\epsilon(x) = \{y \in V : \|x - y\| \leq \epsilon\}$ is compact, since closed balls are closed and bounded.

As alluded to earlier, one of the most appealing features of compact sets is that they interact well with continuous functions. In particular, a continuous function *always* achieves a maximum and minimum value on a compact set. In the following proposition, we summarize some of the most useful properties.

Proposition 1.14 (Compactness & Continuity) *Consider two normed vector spaces, $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$.*

1. *Let $f : V \to W$ be a continuous map. For any compact set $K \subseteq V$, the image of K under f, $f(K) \subseteq W$, is also compact.*
2. *Let $f : V \to \mathbb{R}$ be a continuous map. For any compact set $K \subseteq V$, the set $f(K)$ has a maximum and a minimum value. Consequently,*

$$\sup_{x \in K} f(x) = \max_{x \in K} f(x) \text{ and } \inf_{x \in K} f(x) = \min_{x \in K} f(x). \tag{1.31}$$

Property (2) is particularly useful when bounding the values of a function over a given set. Oftentimes, one can find a compact set containing the given set, and use the maxima and minima on the compact set to bound the function values on the given set. The fact that a continuous function achieves a maximum and a minimum value over a compact set means that there is *no risk* of the function “blowing up” to $+\infty$ or $-\infty$ on the set.

The following exercise presents a nice consequence of this proposition.

Exercise 1.22 Let $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ be normed vector spaces and $f : V \to W$ be a continuous map. Show that on any compact set $K \subseteq V$, the function $g : V \to \mathbb{R}$, defined $g(x) = \|f(x)\|_W$, attains a maximum and a minimum value. *Hint: are norms continuous functions?*

Definition 1.22 offers one definition of continuity of a map between normed vector spaces. What else might we be interested in? Oftentimes, the form of continuity presented in Definition 1.22 is not strong enough! Next, we consider a stronger form of continuity, called Lipschitz continuity.

Definition 1.26 (Lipschitz Continuity) Let $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ be two normed vector spaces. A mapping $f : V \to W$ is said to be Lipschitz continuous on V if there exists a constant $L \geq 0$, called a Lipschitz constant, for which

$$\|f(x) - f(y)\|_W \leq L \|x - y\|_V, \ \forall x, y \in V. \tag{1.32}$$

Remark 1.14 Unlike continuity, we specify Lipschitz continuity as a global property of a function. Lipschitz continuity is defined over the entire space, rather than at a single point.

By nature of the name Lipschitz *continuity*, it's natural to expect a Lipschitz continuous function to be continuous in the sense of Definition 1.22. In the next proposition, we confirm that our expectations are met.

Proposition 1.15 *Consider a function $f : V \to W$ between normed vector spaces $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$. If f is Lipschitz continuous on V, then it is continuous on V.*

Proof Suppose $f : V \to W$ is Lipschitz continuous. We must show that f is continuous at every $x \in V$. First, fix a point $x \in V$ and a number $\epsilon > 0$. To show that f is continuous at x, we must find a number $\delta > 0$ - possibly dependent on ϵ and x, such that $\|x - y\|_V < \delta$ implies $\|f(x) - f(y)\|_W < \epsilon$. In order to identify such a δ, we appeal to the definition of Lipschitz continuity. By Definition 1.26, there exists a constant $L \geq 0$ such that for all $x, y \in V$, $\|f(x) - f(y)\|_W \leq L \|x - y\|_V$. Looking at the inequalities,

$$\|f(x) - f(y)\|_W \leq L \|x - y\|_V \qquad \text{(what we have)} \tag{1.33}$$

$$\|f(x) - f(y)\|_W < \epsilon \qquad \text{(what we want)}, \tag{1.34}$$

it seems reasonable that choosing $\delta = \epsilon / L$ will meet our needs. Let's check that this value works. For any y satisfying $\|x - y\|_W < \delta = \epsilon / L$, we have

$$\|f(x) - f(y)\|_W \leq L \|x - y\|_V < L \cdot \frac{\epsilon}{L} = \epsilon. \tag{1.35}$$

Thus, we conclude that f is continuous at x. Since x was chosen arbitrarily, we conclude that f is continuous on V. □

Above, we showed that all Lipschitz continuous functions are continuous. What about the other direction? Are all continuous functions Lipschitz? As the following exercise illustrates, Lipschitz continuity is a *strictly stronger* condition than continuity.

Exercise 1.23 The function $f : \mathbb{R} \to \mathbb{R}$, defined $f(x) = x^2$ is well-known to be continuous. Show that it is *not* Lipschitz continuous. *Hint: proceed by contradiction.*

An important example of a Lipschitz function on any normed vector space $(V, \|\cdot\|)$ is the norm $\|\cdot\|$ itself, as a function from $V \to \mathbb{R}$.

Proposition 1.16 *For $(V, \|\cdot\|)$ a normed vector space, $\|\cdot\| : V \to \mathbb{R}$ is Lipschitz continuous.*

This result implies that the norm of a normed vector space is a continuous function.

Exercise 1.24 Prove Proposition 1.23.

So far in our discussion of continuity, we've dealt only with *arbitrary* mappings between normed vector spaces. What can we say about the continuity of *linear* transformations? Consider, for example, the simple linear transformation from $\mathbb{R} \to \mathbb{R}$, defined $f(x) = ax$, $a \in \mathbb{R}$. It's clear that this function is continuous; in fact, it is Lipschitz continuous with Lipschitz constant $|a|$. Therefore, as an initial guess, it seems not too far-fetched to suggest that linear transformations between normed vector spaces are Lipschitz continuous.

Unfortunately, this is *not* true for general linear transformations between normed vector spaces! The class of linear transformations for which this *is* true is called the class of *bounded linear operators.*

Definition 1.27 (Bounded Linear Operator/Transformation) Consider two normed vector spaces, $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$, and a linear transformation $A : V \to W$. A is said to be a bounded linear operator/transformation if there exists a $K \geq 0$ for which

$$\|Ax\|_W \leq K \|x\|_V \, , \; \forall x \in V. \tag{1.36}$$

Remark 1.15 Here, we've undergone a little bit of a terminology change from "transformation" to "operator." The "operator" terminology is more commonly used in functional analysis, while "transformation" is more often seen in linear algebra. These two terms are entirely interchangeable - for instance one may say "bounded linear operator" or "bounded linear transformation" in reference to the definition above.

Thus, a linear transformation is bounded if it doesn't "scale" any vector to be arbitrarily large! Taking a closer look at the definition of a bounded linear operator, one immediately notices a similarity to the definition of Lipschitz continuity. This observation leads to the following result.

Proposition 1.17 *Bounded linear operators are Lipschitz continuous.*

Proof Consider a bounded linear operator $A : V \to W$. Such a map satisfies $\|Ax\|_W \leq K \|x\|_V \, , \; \forall x \in V$, where $K \geq 0$ is some fixed constant. By the axioms of a vector space, for all $x, y \in V$, $x - y \in V$. Therefore, one has that for all $x, y \in V$,

$$\|Ax - Ay\|_W = \|A(x - y)\|_W \leq K \|x - y\|_V \, . \tag{1.37}$$

We conclude that A is Lipschitz continuous with Lipschitz constant K. □

One of the most interesting and useful facts about the set of bounded linear operators between two vector spaces is that they themselves form a vector space! In the following theorem, we define the vector space of bounded linear operators.

Theorem 1.2 (Vector Space of Bounded Linear Operators) *Consider two normed vector spaces $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ over $\mathbb{K}$. The set of all bounded linear operators between V and W, denoted $\mathcal{L}(V, W)$, is itself a vector space under the operations $+$ and $(\cdot)$, defined*

$$(A + B)(v) = A(v) + B(v) \tag{1.38}$$

$$(c \cdot A)(v) = c(Av), \tag{1.39}$$

for all $v \in V$ and $c \in \mathbb{K}$.

Exercise 1.25 Prove Theorem 1.2.

One may also verify that the *composition* of any two bounded linear operators is a bounded linear operator.

Proposition 1.18 (Composition of Bounded Linear Operators) *Consider two bounded linear operators, $A \in \mathcal{L}(V, W)$ and $B \in \mathcal{L}(U, V)$. The composition of A and B, denoted AB, is defined $ABv = A(Bv)$. AB is a bounded linear operator from U to W.*

Remark 1.16 The convention of writing the composition $A \circ B$ as AB derives from matrix multiplication, where one writes the product of two matrices A and B as AB.

Now that we've defined a vector space of bounded linear operators, it's natural to seek out a norm that turns this vector space into a normed vector space. A natural choice would be to define a norm on $\mathcal{L}(V, W)$ in terms of the norms on V and W.

Definition 1.28 (Induced Operator Norm) Let $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ be normed vector spaces and $A \in \mathcal{L}(V, W)$ a bounded linear operator from V to W. The induced operator norm of A is defined,

$$\|A\|_{V,W} = \sup_{x \in V \setminus \{0\}} \frac{\|Ax\|_W}{\|x\|_V}. \tag{1.40}$$

Remark 1.17 The "induced" in the name "induced operator norm" refers to the fact that the norm $\|\cdot\|_{V,W}$ is induced by the norms on V and W. For convenience, one often refers to an induced operator norm simply as an "operator norm." If no choice of norm on an operator is specified, it is typically assumed to be the operator norm induced by the normed vector spaces it maps between.

Remark 1.18 The $\sup_{x \in V \setminus \{0\}}$ simply ensures that we don't divide by zero in the quotient defining the operator norm. Shortly, we'll state a few equivalent ways of calculating the operator norm that avoid the use of a quotient.

An important special case of the operator norm is the following.

Definition 1.29 (Left Multiplication Operator/Induced Matrix Norm) Consider a matrix $A \in \mathbb{K}^{m \times n}$, and two normed vector spaces $(\mathbb{K}^m, \|\cdot\|_b)$ and $(\mathbb{K}^n, \|\cdot\|_a)$.

1. Left Multiplication Operator: the linear operator of left multiplication with A, is the operator $L_A : \mathbb{K}^n \to \mathbb{K}^n$, defined $L_A v := Av$ for all $v \in \mathbb{K}^n$.
2. Induced Matrix Norm: the induced matrix norm of A is defined $\|A\|_{a,b} = \|L_A\|_{V,W}$.

Remark 1.19 If the spaces $\mathbb{K}^m$ and $\mathbb{K}^n$ have the same type of norm, for instance the 2-norm, one will simply denote the induced matrix norm as $\|A\|_2$, rather than $\|A\|_{2,2}$. In this case, one refers to the induced 2-norm of A simply as the 2-norm of A.

Example 1.18 (Matrix 2-Norm) The induced 2-norm of $A \in \mathbb{K}^{m \times n}$ (where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$), is calculated $\|A\|_2 = \sigma_{\max}(A)$.

Proposition 1.19 (Operator Norms Define a Normed Vector Space) *Let $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ be normed vector spaces. The space $\mathcal{L}(V, W)$, equipped with the operator norm $\|\cdot\|_{V,W}$, is a normed vector space.*

Exercise 1.26 Prove Proposition 1.19 by showing $\|\cdot\|_{V,W}$ is a norm on $\mathcal{L}(V, W)$.

It's important to note that the operator norm $\|\cdot\|_{V,W}$ is generally not the only norm giving $\mathcal{L}(V, W)$ the structure of a normed vector space. Rather, operator norms are a *natural* choice of norm that derive from the normed vector spaces they map between.

Example 1.19 The Frobenius norm of a matrix $A \in \mathbb{K}^{m \times n}$, $\|A\|_F = \sqrt{\operatorname{tr}(A^* A)}$, is not induced by any ℓ^p-norms on $\mathbb{K}^n$ and $\mathbb{K}^m$.

In addition to endowing the vector space $\mathcal{L}(V, W)$ the structure of a normed vector space, the induced operator norm $\|\cdot\|_{V,W}$ enjoys a number of other useful properties.

Proposition 1.20 (Properties of the Operator Norm) *Consider three normed vector spaces, $(U, \|\cdot\|_U)$, $(V, \|\cdot\|_V)$, and $(W, \|\cdot\|_W)$. The following properties are satisfied:*

1. *Submultiplicative: For all $A \in \mathcal{L}(V, W)$ and $B \in \mathcal{L}(U, V)$, $\|AB\|_{U,W} = \|A\|_{V,W} \|B\|_{V,W}$.*
2. *Vector Inequality: For all $A \in \mathcal{L}(V, W)$ and $x \in V$, $\|Ax\|_W \leq \|A\|_{V,W} \|x\|_V$.*
3. *Equivalent Definitions: The operator norm is equivalently computed by the formulas,*

$$\|A\|_{V,W} = \sup_{\|x\|_V=1} \|Ax\|_W \quad \textit{and} \quad \|A\|_{V,W} = \inf\{K \geq 0 : \|Ax\| \leq K \|x\| \ \forall x \in V\}. \tag{1.41}$$

Let's run through the different components of this proposition. The first item of the proposition tells us that operator norms are submultiplicative - if we take two bounded linear operators and *compose* them, the operator norm of their composition $AB : U \to W$ is bounded above by the product of their operator norms. The second item tells us that the operator norm provides us with an upper bound on how much a linear operator scales the norm of a vector. The final item tells us two equivalent ways of calculating the operator norm. Due to the scaling property of norms, one has

$$\sup_{\|x\|_V=1} \|Ax\|_W = \sup_{x \in V \setminus \{0\}} \frac{\|Ax\|_W}{\|x\|_V}. \tag{1.42}$$

This enables us to calculate the operator norm without the use of a quotient. The second formula for the operator norm - which is not as practical for computation - yields an interpretation of the operator norm as the *tightest possible* Lipschitz constant of the operator A. To complete our study of the space of linear operators, we define some special bounded linear operators.

Definition 1.30 (Identity Operator) Let V be a normed vector space. The identity operator on V, $\mathrm{Id}_V : V \to V$ is defined $\mathrm{Id}_V\, v = v$, for all $v \in V$.

Exercise 1.27 Assuming both the domain and codomain instances of V are equipped with the same norm, confirm that the identity operator is a bounded linear operator.

Note that if $V = \mathbb{R}^n$, the identity linear operator $\mathrm{Id}_{\mathbb{R}^n}$ coincides with the linear operator L_{I_n} of left multiplication by the $n \times n$ identity matrix, I_n.

Definition 1.31 (Invertible Linear Operator) Let V and W be normed vector spaces. A linear operator $A \in \mathcal{L}(V, W)$ is said to be invertible if there exists a $B \in \mathcal{L}(W, V)$ for which $AB = \mathrm{Id}_W$ and $BA = \mathrm{Id}_V$. In this case, B is said to be the *inverse* of A, denoted A^{-1}.

Remark 1.20 Note that we have used the language *the* inverse. One may show that the inverse of a linear operator is always unique - thus, it makes sense to talk about *the* inverse rather than *an* inverse.

Using invertible linear operators, one defines what it means for two vector spaces to be *isomorphic.*

Definition 1.32 (Isomorphism) Two normed vector spaces $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ are said to be isomorphic if there exists an invertible linear mapping $A \in \mathcal{L}(V, W)$. Such an A is said to be an isomorphism of the vector spaces V and W.

Roughly speaking, if two vector spaces are isomorphic, they share the same underlying "algebraic structure." It is a fundamental theorem of linear algebra that every finite-dimensional vector space over $\mathbb{R}$ is isomorphic to $\mathbb{R}^n$ for some n. We conclude that any finite-dimensional vector space over $\mathbb{R}$ has the same *algebraic structure* as $\mathbb{R}^n$, for some n. Consequently, it's often sufficient to prove results for general finite-dimensional vector spaces in $\mathbb{R}^n$.

1.4 Banach Spaces

Thus far in our study of normed vector spaces, we've focused primarily on questions regarding the structure of sets, norms, and mappings of vector spaces. In this section, we study *convergence*, another fundamental subject of basic analysis. In order to fully understand convergence, we'll need a little bit more structure than the basic normed vector space setting. In this section, we develop the basic theory of *Banach spaces*, a special class of normed vector space which enjoys additional convergence properties.

In order to define Banach spaces, we first need to understand sequences in normed vector spaces. Let's start by discussing sequences in $\mathbb{R}$. In $\mathbb{R}$, we think of sequences as ordered lists of real numbers, for example,

$$(a_1, a_2, a_3, ...), \text{ where each } a_i \in \mathbb{R}. \tag{1.43}$$

What such a sequence *really* is is a mapping from $\mathbb{N} \to \mathbb{R}$, taking an index of the sequence in $\mathbb{N}$ and mapping it to a value in $\mathbb{R}$. Above, we map $1 \mapsto a_1$, $2 \mapsto a_2$, and so on. This definition is generalized to the setting of vector spaces.

Definition 1.33 (Sequence) A sequence in a vector space V is a mapping $a : \mathbb{N} \to V$. Individual elements of the sequence are denoted a_n, while the entire sequence object is denoted $\{a_n\} \subseteq V$.

Remark 1.21 The definition of a sequence of a mapping $a : \mathbb{N} \to V$ suggests that we might write elements of a sequence as $a(n)$ rather than a_n. However, as we often like to distinguish sequences from "regular" mappings, we favor the notation a_n over $a(n)$. Likewise, instead of referring to a sequence via the actual mapping $a : \mathbb{N} \to V$, one writes $\{a_n\} \subseteq V$ to define a sequence $a : \mathbb{N} \to V$.

Using the language of normed vector spaces, we may formulate a precise definition for the *convergence* of a sequence.

Definition 1.34 (Sequential Convergence) Consider a normed vector space $(V, \|\cdot\|)$. A sequence $\{a_n\} \subseteq V$ is said to converge if there exists a vector $a \in V$ such that, for all $\epsilon > 0$, there exists an $N \in \mathbb{N}$ (possibly dependent on ϵ, for which $n \geq N \implies \|a_n - a\| < \epsilon$. In this case, one says that a is the limit of $\{a_n\}$, and writes $\lim_{n\to\infty} a_n = a$.

What is this definition saying? Essentially, a sequence $\{a_n\}$ converges to a limit a if a_n eventually comes and remains arbitrarily close to a. Here, ϵ encodes a specification of how close we want the sequence to come to a, and N tells us how far into the sequence we need to look for $\{a_n\}$ to come and remain within a distance ϵ of a.

Since we're working in normed vector spaces, sequences can take on far more interesting forms than simple sequences of real numbers. Sequences of *functions* will be of particular interest to us. In the following example, we focus on sequences in a particularly important function space.

Example 1.20 (Uniform Convergence) Consider the normed vector space $(V, \|\cdot\|_\infty)$, where V is the set of functions $f : \mathbb{R} \to \mathbb{R}$ with finite supremum norm,

$$\|f\|_\infty = \sup_{t\in\mathbb{R}} |f(t)|. \tag{1.44}$$

Sequences in this normed vector space are sequences of *functions*, $\{f_n\} \subseteq V$. If a sequence of functions $\{f_n\}$ converges to a function f with respect to the supremum norm $\|\cdot\|_\infty$, one says that f_n *converges uniformly* to f. Interestingly, if a sequence $\{f_n\}$ of continuously differentiable functions converges uniformly to a function f, the limiting function will also be continuously differentiable, with derivative equal to the limit of the derivatives of the f_n.

Fortunately, sequences interact well with the algebraic operations of a vector space, as well as with continuous functions between normed vector spaces.

Proposition 1.21 (Sequential Limit Properties) *Let $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ be normed vector spaces over $\mathbb{K}$ and $\{a_n\}, \{b_n\} \subseteq V$ convergent sequences with limits a and $b \in V$.*

1. *Algebraic Combinations: for $\alpha, \beta \in \mathbb{K}$, $\lim_{n\to\infty}(\alpha a_n + \beta b_n) = \alpha a + \beta b$.*
2. *Function Composition: for any continuous mapping $f : V \to W$, $\lim_{n\to\infty} f(a_n) = f(a)$.*

So far, we've discussed a definition of convergence which requires a candidate limit in order to certify convergence. Although in simple cases, it's not too hard to come up with a candidate for a limit (e.g. we can guess $1/n \to 0$ without too much trouble), for more complex sequences this becomes challenging.

What we'd like is a way to certify a sequence converges *without* knowing ahead of time what the sequence converges to. Let's reason about how we might do this in $\mathbb{R}$. In $\mathbb{R}$, our intuition tells us that a sequence $\{a_n\}$ will converge if its terms get closer and closer together as n grows large. We generalize this idea to normed vector spaces with the following definition.

Definition 1.35 (Cauchy Sequence) Let $(V, \|\cdot\|)$ be a normed vector space. A sequence $\{a_n\} \subseteq V$ is said to be a Cauchy sequence if, for all $\epsilon > 0$, there exists an $N > 0$ (possibly dependent on ϵ), such that $n, m \geq N \implies \|a_n - a_m\| < \epsilon$.

Remark 1.22 Frequently, instead of saying that a sequence $\{a_n\} \subseteq V$ is a Cauchy sequence, we will simply say "$\{a_n\}$ is Cauchy," and drop the extra label of "sequence."

Based on the definition, we see that a sequence is Cauchy if, given any $\epsilon > 0$, the terms of the sequence eventually come and remain within a distance ϵ of each other. In other words, for large n, the terms of the sequence begin to "cluster" together. Notably, since this definition only relies on the terms of the sequence, one does not require a candidate limit to prove that a given sequence is Cauchy.

Let's determine if Cauchy sequences meet our requirement. Is is true that a Cauchy sequence in any normed vector space converges? In finite dimensions, the answer is *yes*, but in infinite dimensions, the answer is (of course) much more subtle. Since it is not a given that a Cauchy sequence converges in any given normed vector space, we define a special class of normed vector spaces in which Cauchy sequences *always* converge. Although this might initially seem like a restrictive condition, this class of normed vector spaces turns out to be quite vast, containing a wide variety of interesting spaces.

Definition 1.36 (Banach Space) A Banach space is a normed vector space $(V, \|\cdot\|)$ in which every Cauchy sequence converges to a limit in V.

Remark 1.23 Spaces in which Cauchy sequences always converge to a limit in the space are also referred to as *complete spaces* - this terminology extends to more general spaces beyond normed vector spaces. Using this language, one refers to a Banach space is a *complete* normed vector space.

Remark 1.24 It's critical to note that for a space V to be a Banach space, the limits of all Cauchy sequences must belong to V. It's not enough to have a Cauchy sequence converge to a limit outside of the space V - the limit must be contained in V itself.

Since every Cauchy sequence in a Banach space converges, Banach spaces afford us the ability to study the convergence of sequences without actually knowing what the limits of the sequences might be. This is an incredibly powerful ability that has far reaching implications. In fact, one can use the consequences of completeness to prove the existence and uniqueness of solutions to certain differential equations.

Now, we consider some important examples of Banach spaces. As we alluded to above, it is true that *every* finite-dimensional normed vector space is a Banach space.

Theorem 1.3 *Any finite-dimensional normed vector space is a Banach space.*

Proof See Problem 1.3. □

This result alone encompasses an enormous class of interesting spaces. Because of Theorem 1.3, we must turn to infinite-dimensional spaces to find other examples of Banach spaces. A particularly rich class of Banach spaces is supplied by the ℓ^p and L^p spaces, which we now define.

Definition 1.37 (ℓ^p Space) Fix a number $p \in [1, \infty)$. The normed vector space $(\ell^p, \|\cdot\|_{\ell^p})$ has as vectors the set of all sequences $u : \mathbb{N} \to \mathbb{R}$ with finite ℓ^p norm,

$$\|u\|_{\ell^p} = \left(\sum_{n=1}^{\infty} |u_n|^p \right)^{\frac{1}{p}} < \infty. \tag{1.45}$$

This space is equipped with the operations of addition and scalar multiplication of sequences. For $p = \infty$, the normed vector space $(\ell^\infty, \|\cdot\|_{\ell^\infty})$ consists of all sequences $u : \mathbb{N} \to \mathbb{R}$ with finite ℓ^∞ norm,

$$\|u\|_{\ell^\infty} = \sup_{n \in \mathbb{N}} |u_n| < \infty. \tag{1.46}$$

Remark 1.25 The definition of an ℓ^p space can be easily extended from sequences $u : \mathbb{N} \to \mathbb{R}$ to sequences with other domains and codomains, for instance $u : \mathbb{Z}_{\geq 0} \to \mathbb{R}$ or $u : \mathbb{N} \to \mathbb{R}^m$. In these two examples, one would adjust the indices of the sum or trade the absolute value for a norm on $\mathbb{R}^m$ to accommodate the different domain or codomain of u. We'll return to the general case in Chapter 3.

As one can observe from the definition, the vectors of an ℓ^p space are *infinite sequences* of real numbers with bounded ℓ^p norm.

Definition 1.38 (L^p Space) Fix a number $p \in [1, \infty)$. The normed vector space $(L^p, \|\cdot\|_{L^p})$ consists of all functions $f : \mathbb{R} \to \mathbb{R}$ with finite L^p norm,

$$\|f\|_{L^p} = \left(\int_{\mathbb{R}} |f(x)|^p dx \right)^{\frac{1}{p}} < \infty \tag{1.47}$$

This space is equipped with the operations of function addition and scalar multiplication. For $p = \infty$, the normed vector space $(L^\infty, \|\cdot\|_{L^\infty})$ consists of all functions $f : \mathbb{R} \to \mathbb{R}$ with finite L^∞ norm,

$$\|f\|_{L^\infty} = \sup_{x \in \mathbb{R}} |f(x)| < \infty. \tag{1.48}$$

As opposed to ℓ^p spaces, where elements are real-valued sequences, in an L^p space, the elements are real-valued *functions* with bounded L^p norm. Despite the change from sequence to function, the ℓ^p and L^p norms have clear parallels in their definitions.

Remark 1.26 As with ℓ^p spaces, the definition of L^p spaces provided above is easily extended to more general domains and codomains (for example to functions taking values in $\mathbb{R}^n$). As with ℓ^p, we'll return to the general case in Chapter 3.

Remark 1.27 In defining L^p spaces as spaces of functions with finite integrals for $p \in [1, \infty)$, we've glossed over a number of concerns regarding Riemann/Lebesgue integrability of the functions f. Since properly treating these concerns requires a (significant & not immediately revealing) detour into a measure theory, we simply assume that elements of L^p spaces are "well-behaved" enough to have well-defined integrals. We also ignore the concerns that follow from functions differing on sets of measure zero having the same norm. We direct the interested reader to the additional reading specified at the end of the chapter for a rigorous treatment of these spaces.

In the previous section, we promised that we would return to finish the story of compactness once we defined Banach spaces. Now, we make good on this promise and provide a characterization of compactness in Banach space through convergent sequences. In order to state this characterization, we first require the concept of a *subsequence*. Given a sequence,

$$a_1, a_2, ..., a_n, ... \tag{1.49}$$

in a vector space V, a subsequence is a subset of the sequence in which elements are picked out in an order that respects that of the original sequence. For instance, a valid subsequence of the sequence above would be,

$$a_1, a_3, a_5, ... \tag{1.50}$$

since the terms of the subsequence appear in the same order as the original sequence. An *invalid* subsequence would then be,

$$a_2, a_1, a_3, a_2, ..., \tag{1.51}$$

which does *not* form a valid subsequence since the terms appear out of order compared to the original sequence. We abstractly define this "preservation of order" requirement in the following definition of a subsequence.

Definition 1.39 (Subsequence) Let V be a vector space and $\{a_n\} \subseteq V$ a sequence. A subsequence of $\{a_n\}$ is a subset $\{a_{n_k}\} \subseteq \{a_n\}$, where $n_k : \mathbb{N} \to \mathbb{N}$ is a strictly increasing sequence of indices,

$$n_1 < n_2 < n_3 < ..., \tag{1.52}$$

which specify the indices of the terms drawn from $\{a_n\}$.

For instance, for the subsequence $a_1, a_3, a_5, ...$, one would define n_k by $n_k = 2k - 1$. This produces the subsequence,

$$a_{n_1} = a_1,\ a_{n_2} = a_3,\ a_{n_3} = a_5, ..., \tag{1.53}$$

which is exactly the desired subsequence. Clearly, the indices of the subsequence are strictly increasing. Using the language of subsequences, one may formulate an equivalent definition of compactness in Banach spaces. Consider the following theorem, the proof of which is beyond our scope.

Theorem 1.4 (Compactness in Banach Space) *Let $(V, \|\cdot\|)$ be a Banach space and $K \subseteq V$. The following provide two equivalent characterizations of the compactness of K:*

1. *K is compact iff every sequence $\{a_n\} \subseteq K$ has a subsequence with a limit in K.*
2. *K is compact iff every sequence $\{a_n\} \subseteq K$ has a subsequence that is Cauchy.*

Remark 1.28 The abbreviation *"iff"* is commonly used as shorthand for "if and only if."

Thus, in Banach spaces, one can detect the compactness of a set K knowing only information about the sequences contained in K.

Exercise 1.28 Using Theorem 1.4, prove that a single-element subset $\{v\}$ of a Banach space (called a *singleton set*) is compact.

1.5 A Refresher on ODEs

Ordinary differential equations (ODEs) are the lingua franca of control in continuous time. In this section, we review some basic properties of scalar, linear differential equations. We postpone a formal discussion of the existence and uniqueness of solutions to such equations until Chapter 2, and only touch upon the most essential conceptual aspects here. As such, our treatment is significantly more informal than the previous sections of this chapter. For the reader concerned by the appalling lack of theorems - don't worry, plenty are coming down the pipeline - sit tight for a few more pages!

We begin by discussing initial value problems. Here, we'll keep our discussion to the case of ordinary differential equations in $\mathbb{R}$. Let $f : \mathbb{R} \to \mathbb{R}$ be a scalar function. The ordinary differential equation (ODE) governed by f is the equation,

$$\frac{d}{dt}x(t) = f(x(t)). \tag{1.54}$$

To find the *general solution* to the ordinary differential equation, one must identify all functions of the form $x : I \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is a nonempty open interval, satisfying $\frac{d}{dt}x(t) = f(x(t))$, for all $t \in I$.

Since it's a bit cumbersome to repeatedly write $\frac{d}{dt}$ one writes $\frac{d}{dt}x(t)$ in shorthand as $\dot{x}(t)$. Additionally, it's common to suppress the argument of $x(t)$. We therefore write the ordinary differential equation in shorthand as,

$$\dot{x} = f(x). \tag{1.55}$$

Now, we consider an *initial value problem* associated to an ordinary differential equation. Given a constant $x_0 \in \mathbb{R}$, the goal of the initial value problem is to solve the problem,

$$\frac{d}{dt}x(t) = f(x(t)),\ x(0) = x_0 \in \mathbb{R}. \tag{1.56}$$

That is, one wishes to find a curve $x : I \to \mathbb{R}$ (for I a nonempty, open interval), where $0 \in I$, satisfying $\frac{d}{dt}x(t) = f(x(t))$, for all $t \in I$ and $x(0) = x_0$. In our shorthand introduced above, one would abbreviate this problem as

$$\dot{x} = f(x),\ x(0) = x_0. \tag{1.57}$$

For a general nonlinear function f, initial value problems are challenging, and at worst impossible, to solve. A special case which has a well-defined solution is the scalar, linear initial value problem.

Theorem 1.5 (Scalar, Linear, First Order IVP) *Let $a \in \mathbb{R}$ be a fixed scalar. Consider the initial value problem,*

$$\dot{x} = ax,\ x(0) = x_0, \tag{1.58}$$

where $x_0 \in \mathbb{R}$ is a fixed initial condition. The unique solution $x : \mathbb{R} \to \mathbb{R}$ to this initial value problem is $x(t) = e^{at}x_0$.

Exercise 1.29 Verify that the solution proposed in Theorem 1.5 solves the initial value problem. You do not need to prove uniqueness - we'll return to this in Chapter 2.

Interestingly, this result states that, not only is $x(t) = e^{at}x_0$ a solution to the proposed initial value problem, but it is the *only* solution! That is, there is *no other function* satisfying the initial value problem. Further, the solution is defined on *all* of $\mathbb{R}$, rather than on some bounded, open interval. These properties are *not* shared by every initial value problem.

Example 1.21 Consider the initial value problem, $\dot{x} = 2\sqrt{|x|}$, $x(0) = 0$. For all $a \geq 0$, the function

$$x(t) = \begin{cases} (t-a)^2 & t \geq a \\ 0 & t < a, \end{cases} \tag{1.59}$$

is a solution of the initial value problem. Thus, the initial value problem has an *infinite* number of solutions.

Thus far, we've only considered *first order*, scalar ordinary differential equations. We can easily extend the definition of a differential equation to one that involves higher derivatives. In the following setup, we denote by $x^{(n)}(t)$ the n'th derivative $\frac{d^n x(t)}{dt^n}$. Let $f : \mathbb{R}^n \to \mathbb{R}$, and consider the ordinary differential equation,

$$x^{(n)}(t) = f(x(t), x^{(1)}(t), ..., x^{(n-1)}(t)). \tag{1.60}$$

Here, a solution of the differential equation is a function $x : I \to \mathbb{R}$ whose n'th derivative equals the function f of its first $n-1$ derivatives (by convention, the 0'th derivative of $x(t)$ is taken to be $x(t)$ itself).

Do we need to construct an entirely new theory for higher order differential equations? Fortunately, by lifting our problem from a *scalar* differential equation to a *vector* differential equation, we can transform any n'th order scalar differential equation into a system of n first order differential equations. Consider the following change of variables for the n'th order initial value problem specified above. Define,

$$\begin{align}
q_0(t) &= x(t) \tag{1.61}\\
q_1(t) &= \dot{x}(t) \tag{1.62}\\
q_2(t) &= \ddot{x}(t) \tag{1.63}\\
&\vdots \tag{1.64}\\
q_{n-1}(t) &= x^{(n-1)}(t). \tag{1.65}
\end{align}$$

Differentiating each of the q variables, one has

$$\begin{align}
\dot{q}_0(t) &= \dot{x}(t) = q_1(t) \tag{1.66}\\
\dot{q}_1(t) &= \ddot{x}(t) = q_2(t) \tag{1.67}\\
\dot{q}_2(t) &= x^{(3)}(t) = q_3(t) \tag{1.68}\\
&\vdots \tag{1.69}\\
\dot{q}_{n-1}(t) &= x^{(n)}(t) = f(x(t), ...x^{(n-1)}(t)). \tag{1.70}
\end{align}$$

We recognize that we can rewrite $f(x(t), ..., x^{(n-1)}(t))$ as $f(q_0(t), ..., q_{n-1}(t))$. Thus, the differential equation $x^{(n)}(t) = f(x(t), ..., x^{(n)}(t))$ can be rewritten as a *vector* differential equation in $\mathbb{R}^n$,

$$\frac{d}{dt}\begin{bmatrix} q_0(t) \\ \vdots \\ q_{n-1}(t) \end{bmatrix} = \begin{bmatrix} q_1(t) \\ \vdots \\ f(q_0(t), ..., q_{n-1}(t)) \end{bmatrix}. \tag{1.71}$$

Defining a vector $q = (q_0, ..., q_{n-1}) \in \mathbb{R}^n$ and a function $F : \mathbb{R}^n \to \mathbb{R}^n$ as $F(q) = (q_1, ..., f(q_0, ..., q_{n-1}))$, we compactly rewrite this differential equation in vector form as,

$$\dot{q} = F(q). \tag{1.72}$$

To solve the differential equation, we must identify a *vector* function $q : I \to \mathbb{R}^n$ satisfying $\dot{q}(t) = F(q(t))$ for all $t \in I$. To recover the solution to our original, scalar ODE, we the extract the component function $q_0(t)$ of $q(t)$, which by definition equals the solution $x(t)$ to the original ODE.

Initial value problems are also defined similarly to the scalar case. In order to define an initial value problem, one specifies a *vector* initial condition, $q_0 \in \mathbb{R}^n$, to get the problem,

$$\dot{q} = F(q),\ q(0) = q_0. \tag{1.73}$$

Just as in the scalar case, the solution to the vector initial value problem is a function $q : I \to \mathbb{R}^n$ satisfying $\dot{q}(t) = F(q(t))$ for all $t \in I$ and $q(0) = q_0$. The transformation from an n'th order scalar ODE into a first order vector ODE tells us that it's sufficient just to develop a theory for first order, vector ODEs to study general ODEs. We resume this story in the next chapter!

1.6 Further Reading

For an abstract treatment of linear algebra, we refer the reader to the texts *Linear Algebra* by Friedberg, Insel, & Spence [13], and *Linear Algebra Done Right* by Sheldon Axler [5]. For a user-friendly introduction to real analysis in $\mathbb{R}$, we recommend *Understanding Analysis* by Stephen Abbott [1]. For a similarly user-friendly treatment of analysis in normed vector spaces & Banach spaces, we refer the reader to *Measure, Integration, & Real Analysis* by Sheldon Axler [4]. Here, the reader can find proofs of a number of the concepts treated in this section, as well as a rigorous treatment of L^p spaces, enabled by measure theory.

1.7 Problems

Problem 1.1 (Consequences of Norm Equivalence) In this problem, we'll consider some simple consequences of norm equivalence. Consider a vector space V with two equivalent norms, $\|\cdot\|_a$ and $\|\cdot\|_b$.

1. Show that a sequence $\{v_n\} \subseteq V$ converges with respect to $\|\cdot\|_a$ if and only if it converges with respect to $\|\cdot\|_b$.
2. Show that a mapping $f : V \to V$ is continuous with respect to norm $\|\cdot\|_a$ if and only if it is continuous with respect to $\|\cdot\|_b$. Does the same property hold for Lipschitz continuity?

Problem 1.2 (Unbounded Linear Operators) We know that every linear transformation between finite-dimensional normed vector spaces is bounded. In infinite dimensions, we're not quite so lucky! Product an example of an unbounded linear transformation from $\ell^2 \to \ell^2$. *Hint: The ℓ^2 norm is defined as an infinite series - think about some series that converge, and how they can be linear modified to no longer converge. It helps to work with the square of the ℓ^2 norm here.*

Problem 1.3 ($(\mathbb{R}^n, \|\cdot\|)$ is a Banach Space) One may show that in $\mathbb{R}$, a sequence converges (with respect to the absolute value norm) if and only if it is Cauchy. That is, $(\mathbb{R}, |\cdot|)$ is a Banach space. In this problem, we'll show that for any norm $\|\cdot\|$ on $\mathbb{R}^n$, $(\mathbb{R}^n, \|\cdot\|)$ is a Banach space - this is a special case of the result that *every* finite-dimensional normed vector space is Banach.

1. Consider a sequence $\{v_k\} \subseteq \mathbb{R}^n$. Let $\{v_k^i\} \subseteq \mathbb{R}$ represent the sequence formed from the i'th components of each $v_k \in \mathbb{R}^n$ (i.e. $v_k = (v_k^1, v_k^2, ..., v_k^n)$). Show that the sequence $\{v_k\}$ converges to a vector $v \in \mathbb{R}^n$ with respect to the ℓ^∞ norm on $\mathbb{R}^n$ if and only if each component sequence $\{v_k^i\}$ converges to v^i in $\mathbb{R}$.

2. Show that $\{v_k\} \subseteq \mathbb{R}^n$ is a Cauchy sequence with respect to the ℓ^∞ norm on $\mathbb{R}^n$ if and only if each component sequence $\{v_k^i\}$ is Cauchy in $\mathbb{R}$.
3. Using norm equivalence on $\mathbb{R}^n$, show that for any norm on $\mathbb{R}^n$, a sequence is Cauchy if and only if it is convergent.

Problem 1.4 (The Space of Polynomials) Consider the set $\mathcal{P}$ consisting of all polynomials (of all finite degrees) $p : I \to \mathbb{R}$ on a compact interval $I \subseteq \mathbb{R}$.

1. Show that $(\mathcal{P}, \|\cdot\|_\infty)$, where $\|\cdot\|_\infty$ is the sup norm, $\|p\|_\infty = \sup_{t \in I} |p(t)|$, is a normed vector space.
2. Is $(\mathcal{P}, \|\cdot\|_\infty)$ a Banach space? Explain why or why not. *Hint: think about Taylor series.*

Problem 1.5 (Systems of First Order Equations)

1. Show that an n'th order linear ODE,

$$\frac{d^n x}{dt^n} + a_{n-1}\frac{d^{n-1}x}{dt^n} + ... + a_1 \frac{dx}{dt} + a_0 x = 0,\ a_i \in \mathbb{R}, \tag{1.74}$$

can be rewritten as a system of n, first order differential equations of the form,

$$\dot{z} = Az, \tag{1.75}$$

where $z \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. This tells us that it's sufficient to examine *linear systems of first order ODEs* in order to reach conclusions about linear n'th order ODEs.

2. Show that an n'th order recurrence,

$$x[k+n] + a_{n-1}x[k+n-1] + ... + a_1 x[k+1] + a_0 x[k] = 0, \tag{1.76}$$

can be rewritten as a system of n, first order recurrences of the form,

$$z[k+1] = Az[k], \tag{1.77}$$

where $z \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. This tells us that it's sufficient to examine *linear systems of first order recurrences* in order to reach conclusions about linear n'th order recurrences.

Problem 1.6 (The Structured Singular Value ★★) The (complex) structured singular value is a function from the set of $n \times n$ complex matrices to the reals that helps us understand the "gain" of matrices with structured uncertainty. The first step towards defining the structured singular value is to define a set of matrices $\underline{\Delta} \subseteq \mathbb{C}^{n \times n}$. Let $r_1, ..., r_S$ and $m_1, ..., m_F$ be positive integers for which $\sum_{i=1}^{S} r_i + \sum_{j=1}^{F} m_j = n$. Then, define a set $\underline{\Delta} \subseteq \mathbb{C}^{n \times n}$ as

$$\underline{\Delta} := \{\mathrm{blkdiag}(\delta_1 I_{r_1}, ..., \delta_S I_{r_S}, \Delta_{S+1}, ..., \Delta_{S+F}) : \delta_i \in \mathbb{C},\ \Delta_{s+j} \in \mathbb{C}^{m_j \times m_j}\}, \tag{1.78}$$

where I_k represents the $k \times k$ identity matrix. In short, $\underline{\Delta}$ is the set of block diagonal matrices with *repeated scalar blocks* of dimensions $r_i \times r_i$ (these are the blocks $\delta_i I_{r_i}$) and and *full blocks* of dimensions $m_j \times m_j$ (these are the blocks Δ_{S+j}). Given a matrix $M \in \mathbb{C}^{n \times n}$ and a set $\underline{\Delta} \subseteq \mathbb{C}^{n \times n}$ of the form above, one defines the structured singular value of M, $\mu_{\underline{\Delta}}(M)$, as follows.

Definition 1.40 (Structured Singular Value) For $M \in \mathbb{C}^{n \times n}$, $\mu_{\underline{\Delta}}(M)$ is defined,

$$\mu_{\underline{\Delta}}(M) := \frac{1}{\inf\{\overline{\sigma}(\Delta) : \Delta \in \underline{\Delta} \text{ and } \det(I - M\Delta) = 0\}}, \tag{1.79}$$

unless no $\Delta \in \underline{\Delta}$ makes $I - M\Delta$ singular, in which case $\mu_{\underline{\Delta}}(M) := 0$.

Note that here, we use $\overline{\sigma}(M)$ to denote the maximum singular value of M. Based on this definition, $\mu_{\underline{\Delta}}(M)$ depends both on M and on the set $\underline{\Delta}$. Now, let's get started on our analysis of $\mu_{\underline{\Delta}}$! *Note: In the following problems, you can assume for simplicity that one does not encounter the case where no Δ makes $I - M\Delta$ singular.*

1. Compute $\mu_{\underline{\Delta}}(M)$ in the case where $\underline{\Delta}$ is *unstructured*, i.e. $\underline{\Delta} = \mathbb{C}^{n\times n}$.
2. Recall that the spectral radius of a matrix $M \in \mathbb{C}^{n\times n}$ is defined,

   $$\rho(M) := \max_i |\lambda_i(M)|. \tag{1.80}$$

   Define the set $B_{\underline{\Delta}} = \{\Delta \in \underline{\Delta} : \overline{\sigma}(\Delta) \leq 1\}$. Prove that the structured singular value can be calculated as the following function of spectral radius:

   $$\mu_{\underline{\Delta}}(M) = \sup_{\Delta \in B_{\underline{\Delta}}} \rho(\Delta M). \tag{1.81}$$

   Now, consider the special case where $\underline{\Delta} = \{\delta I_n : \delta \in \mathbb{C}\}$. In this case, show that $\mu_{\underline{\Delta}}(M) = \rho(M)$.
3. Let's consider some additional methods of computing μ. Define the following subset of $\mathbb{C}^{n\times n}$:

   $$\underline{D} = \{\text{blkdiag}(D_1, ..., D_S, d_{S+1}I_{m_1}, ..., d_{S+F}I_{m_F} : D_i \in \mathbb{C}^{r_i\times r_i}, D_i \succ 0, d_{S+j} \in \mathbb{R}_{>0}\}. \tag{1.82}$$

   Prove that, for all $D \in \underline{D}$,

   $$\mu_{\underline{\Delta}}(M) = \mu_{\underline{\Delta}}(D^{\frac{1}{2}} M D^{-\frac{1}{2}}). \tag{1.83}$$

   Then, show that,

   $$\mu_{\underline{\Delta}}(M) \leq \inf_{D\in\underline{D}} \overline{\sigma}(D^{\frac{1}{2}} M D^{-\frac{1}{2}}). \tag{1.84}$$

4. Fix a matrix $M \in \mathbb{C}^{n\times n}$. For the set $\underline{D}$ introduced in part (3), show that the following set is convex for each fixed $\beta \in \mathbb{R}$:

   $$\{D \in \underline{D} : \overline{\sigma}(D^{\frac{1}{2}} M D^{-\frac{1}{2}}) \leq \beta\}. \tag{1.85}$$

   Hint: Rewrite as a linear matrix inequality. Such inequalities are amenable to implementation in convex optimization solvers!

Chapter 2
Linear Dynamical Systems

In this chapter, we begin in earnest our study of systems and control theory. First, we introduce the main players in linear systems theory: linear dynamical systems and their state space representations. Following this, we study solutions to state space representations of linear systems, and see how the state space representations generate formal linear dynamical systems. Then, we'll move on to study linear systems from the I/O perspective, developing the theory of Laplace and $\mathcal{Z}$-transforms along the way. Let's begin!

2.1 Dynamical Systems & State Space Models

In order to develop a precise, mathematical theory of systems and control, it's vital that we understand what control systems actually are. Let's begin with a simple, motivating example. Suppose I have a shower with a handle I can use to control the water temperature. If I stick my hand into the water, I can gain some information as to whether the temperature is too high or too low, and I can adjust the position of the handle accordingly. I can continue to repeat this process - move handle, touch water, correct handle position - until the shower is at a temperature I want. If I continually take measurements and adjust the handle position, I can also adapt to unanticipated changes in the environment, such as my roommates washing dishes and reducing the supply of hot water. In other words, by adapting my actions according to measurements, I can become *robust* to changes in the shower environment.

Fundamentally, this is an example of a *feedback control system*, a system with an input (the position of the handle), a measurement (the temperature of the water that I estimate with my hand), a state (the *true* temperature of the water), internal dynamics (how the handle position affects the water temperature), and a description of time (the number of seconds that have passed). By incorporating feedback from the environment into my actions, I can *control* the system to reach a desired temperature.

This is the fundamental idea of feedback control: by measuring the environment, we can make informed decisions that enable us to control the state of the environment. Additionally, by taking repeated measurements, we can make decisions that adapt to unexpected events that might occur. Thus, in addition to affording us the ability of control over our environment, feedback yields the potential to be *robust* to uncertainty and disturbances in the environment.

2.1.1 Causal Input/Output Dynamical Systems

In order to develop the mathematical foundations of systems and control theory, we first need a precise definition for a *system*. Let's distill the most essential components of the shower system to gain some insight into the problem of determining a precise definition for an abstract, mathematical system. First, let's focus on the different objects making up the shower system. For simplicity, we'll assume that we have an infinite supply of hot water, and that the temperature of the shower is entirely determined by its current temperature, the time that has passed, and the history of shower handle positions. The key objects of this simple shower system are the following.

1. Time: we know the time of day at which we entered the shower, and the current time of day. We can represent both the entry time and the current time with a real number, $t \in \mathbb{R}$, corresponding to the number of seconds (or any other appropriate unit of time) that have passed since the beginning of the day.
2. Inputs: the shower handle was an *input* to the shower system. Input signals to the system composed of different positions of the shower handle over time. We can measure the values of inputs by the angle, $\theta(t) \in \mathbb{R}$, of the shower handle at time t. An input signal would therefore be a function of time, $\theta(\cdot) : \mathbb{R} \to \mathbb{R}$, assigning to each time a shower handle position.
3. Outputs: the outputs (measurements) we took of the shower system were tests of the water temperature with our hand. Since we only took measurements with our hand, and not a thermometer, the measurements of our system might take on values in a set,

$$\{\text{icy, cold, mild, hot, ouch!}\} \tag{2.1}$$

   Measurement *signals* would then be mappings from time, $t \in \mathbb{R}$, to this set of measurement values.
4. State: we know that our set of measurements, {icy, cold, mild, hot, ouch!} doesn't quite cover the actual temperature of the system! We define the *state* of the shower to be the *actual* water temperature, $T \in \mathbb{R}$, measured in degrees Celsius (or any other appropriate unit of temperature). This describes the entire state of the shower at a given time t. Using knowledge of the state, T, time, t, and input, we should be able to *completely determine* what the shower will do next (within the scope of our very simple shower model).

How are all of these basic objects tied together? Underneath the shower system, we know there exist some *shower dynamics* that determine how the temperature of the shower changes according to the passage of time and the history of shower handle inputs. Additionally, we know there is some underlying map that determines which measurement value out of the set {icy, cold, mild, hot, ouch!} we will feel given any true temperature and time. These concepts are encoded in the following two maps.

1. State Transition Map: the state transition map of the shower determines how the state (true temperature) of the shower is influenced by the start time, current time, starting temperature of the water, and history of shower handle positions. This gives us a *complete description* of how the true temperature of the shower changes over time. Notably, the current state of the system only depends on the previous and current shower handle positions! The state does not depend on the future inputs to the system (our shower is unfortunately not fancy enough to predict the future).

2. Readout Map: given any time, temperature, and input value, we should know what measurement value our hand is feeling. The readout map maps from a pair of time, temperature, and shower handle position to this measurement value. Notice that the readout map is *memoryless* - it does not require a history of temperatures or a full input signal, only the current temperature and the current input value!

Finally, we know there are a couple of simple properties all of these objects should obey.

1. Time: we can add and subtract time in the shower system without any confusion.
2. Restriction: suppose we have two input signals to the system which match from times t_0 to t_1. Over the time period t_0 to t_1, these two input signals should produce the same behavior, regardless of if they differ after time t_1.
3. Composition: suppose over the time period t_0 to t_1, an input signal takes us from temperature T_0 to temperature T_1, and over the time period t_1 to t_2, an input signal takes us from temperature T_1 to temperature T_2. Then, applying the input signals from t_0 to t_2 should take us from T_1 to T_2.
4. Identity: if no time passes, the temperature of the shower should stay the same.

When stated in context of the shower example, these three conditions are all fairly "obvious." Although it may seem like these points are too trivial to mention, they'll help us make a well-posed, abstract definition of a system that behaves in the way we expect.

This exploratory example provides us with a template definition for a formal *causal input/output dynamical system.* In order to state the formal definition, all we need to do is abstract away the details of the shower into the language of mathematics. As you're reading the definition, relate the formal mathematical expressions to the analogous components of the shower system we outlined above.

Note that, in our formal definition, everything is named similarly to the shower example with the one exception: the "composition" property has been given the shiny new name of the "semigroup axiom" - we'll discuss the rationale behind this after stating the definition.

Definition 2.1 (Causal Input/Output Dynamical System) Let $\mathcal{T} \subseteq \mathbb{R}$ be a nonempty set. A causal, input/output dynamical system on $\mathcal{T}$ is a tuple $\mathcal{D} = (\mathcal{U}, \mathcal{Y}, \Sigma, \varphi, r)$. Each term is defined as follows:

1. Time set: $\mathcal{T}$ is the time set, a subset of $\mathbb{R}$ describing the possible times in the system.
2. Input space: $\mathcal{U}$ is the input space, a set of mappings from $\mathcal{T}$ to a fixed set U,
$$\mathcal{U} \subseteq \{u : \mathcal{T} \to U\}. \tag{2.2}$$
U is referred to as the input value space. Elements of $\mathcal{U}$ represent the possible input signals to the system.
3. Output space: $\mathcal{Y}$ is the output space, a set of mappings from $\mathcal{T}$ to a fixed set Y,
$$\mathcal{Y} \subseteq \{y : \mathcal{T} \to Y\}. \tag{2.3}$$
Y is referred to as the output value space. Elements of $\mathcal{Y}$ represent the possible output (measurement) signals of the system.
4. State space: Σ is the state space, a set representing the possible states of the system.
5. State transition map: φ is the state transition map, a map,
$$\varphi : \mathbf{T} \times \Sigma \times \mathcal{U} \to \Sigma, \tag{2.4}$$

where $\mathbf{T} = \{(t_1, t_0) \in \mathcal{T} \times \mathcal{T} : t_1 \geq t_0\}$, which describes how the state of the system evolves. In particular, for $t_0, t_1 \in \mathcal{T}$ with $t_0 \leq t_1$, $x_0 \in \varSigma$, and $u(\cdot) \in \mathcal{U}$, $\varphi(t_1, t_0, x_0, u(\cdot))$ returns the state of the system at time t_1 after starting from x_0 at time t_0 and applying input signal $u(\cdot)$.

6. Readout map: r is the readout map, a map,

$$r : \mathcal{T} \times \varSigma \times U \to Y, \tag{2.5}$$

which returns the measured output at time t given the system has a current state of $x(t) \in \varSigma$ and a current input value of $u(t) \in U$.

A dynamical system $\mathcal{D}$ must additionally satisfy the following four axioms:

1. Time axiom: for all $t_1, t_2 \in \mathcal{T}$, $t_1 + t_2 \in \mathcal{T}$ and $t_1 - t_2 \in \mathcal{T}$.
2. Restriction axiom: for all $t_0, t_1 \in \mathcal{T}$ with $t_0 \leq t_1$, $x_0 \in \varSigma$, and $u_1(\cdot), u_2(\cdot) \in \mathcal{U}$, one has that

$$u_1(t) = u_2(t)\ \forall t \in [t_0, t_1] \cap \mathcal{T} \implies \varphi(t_1, t_0, x_0, u_1(\cdot)) = \varphi(t_1, t_0, x_0, \tilde{u}_2(\cdot)). \tag{2.6}$$

3. Semigroup axiom: for all $t_0, t_1, t_2 \in \mathcal{T}$ with $t_0 \leq t_1 \leq t_2$, $x_0 \in \varSigma$, and $u(\cdot) \in \mathcal{U}$,

$$\varphi(t_2, t_1, \varphi(t_1, t_0, x_0, u(\cdot)), u(\cdot)) = \varphi(t_2, t_0, x_0, u(\cdot)). \tag{2.7}$$

4. Identity axiom: for all $t \in \mathcal{T}$, $x \in \varSigma$, and $u(\cdot) \in \mathcal{U}$,

$$\varphi(t, t, x, u(\cdot)) = x. \tag{2.8}$$

Phew, what a mouthful! Let's highlight some subtle yet important consequences of the definition.

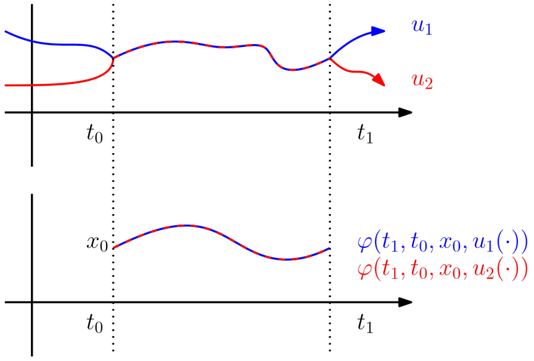
u_1
u_2
t_0
t_1
x_0
$\varphi(t_1, t_0, x_0, u_1(\cdot))$
$\varphi(t_1, t_0, x_0, u_2(\cdot))$
t_0
t_1

Fig. 2.1 The restriction axiom states that if two input signals are equal on a time interval, they will produce the same state behavior on that time interval.

Remark 2.1 (Signal versus Value) In Definition 2.1, we define the input space $\mathcal{U}$ and the output space $\mathcal{Y}$ to be spaces of *signals*, not spaces of values! Associated to the space of input signals $\mathcal{U}$, we have the space of input values, U. Likewise, associated to the space of output signals $\mathcal{Y}$, we have the space of output values, Y. To make this concrete, the space of input

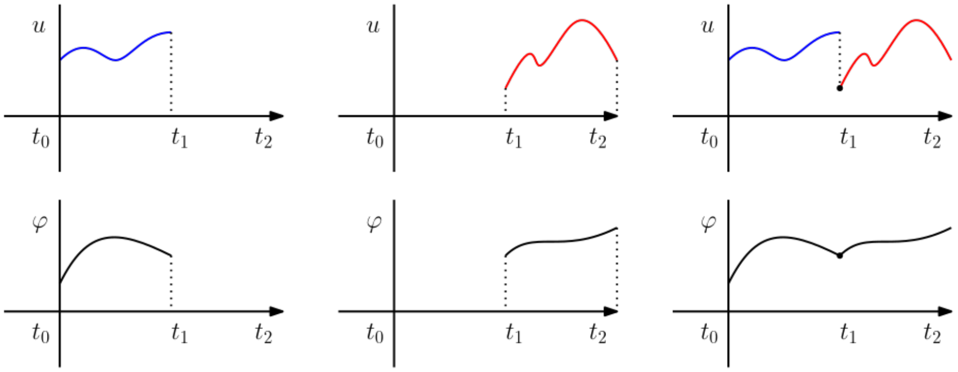

Fig. 2.2 The semigroup axiom states that the state transition map is well-behaved under *composition*. The semigroup axiom implies that, if we stitch together two inputs, the resulting behavior is the same as that which results from applying the first input over its domain and the second input over its domain.

values might be $U = \mathbb{R}^m$, but the space of input signals might be the set of continuous maps from $\mathbb{R} \to \mathbb{R}^m$. To distinguish between the two, we will write $u(\cdot)$ to represent a signal and $u(t)$ to represent its value at time t.

Remark 2.2 (Causality) A system is said to be *causal* if its state and output behavior depends only on its previous and current inputs, *not* on its future inputs! As we can see from the definition of the state transition map, causality is *baked into* the definition of a causal dynamical system. The restriction axiom tells us that the state at time t_1 *only* depends on the state x_0 at time t_0 and the input signal $u(\cdot)$ applied from t_0 to t_1. Thus, any values of the input signal after time t_1 are entirely irrelevant to the behavior of the system. We conclude that any dynamical system satisfying Definition 2.1 must be causal.

Remark 2.3 (Time Axiom) The time axiom can also be stated in terms of *algebraic* language. One may equivalently require $\mathcal{T}$ to be a *subgroup* of $(\mathbb{R}, +)$, the group of real numbers with the addition operation. If you're unfamiliar familiar with algebraic language, this connection isn't something you need to worry about - there are no practical differences between this and what we stated in Definition 2.1.

Remark 2.4 (Semigroup Axiom) Why rename the "composition" axiom as the semigroup axiom? The name *semigroup* alludes to another connection between causal I/O dynamical systems and abstract algebra. If you're interested, read the definition of a semigroup and a semigroup action and see if you can draw a connection between a algebraic semigroups/semigroup actions and the semigroup axiom of Definition 2.1.

Remark 2.5 (Generality) This is but one of many different definitions of an I/O dynamical system, and is by no means the most general definition possible. For instance, the systems proposed above are causal, deterministic (not random), and have fixed input and readout spaces that do not change with state or time. Further, the state transition map is defined on the entire time set, rather than on subsets thereof. Although these assumptions are sufficiently general for our purposes in this course, one should keep these limitations in mind! We direct the reader to the references at the end of the chapter for the more general definitions.

To get some practice with identifying the components of a dynamical system, try the following three exercises. Make sure to state your assumptions where necessary; in each case, you'll need to lay out some simplifying assumptions in order to come up with a manageable dynamical system.

Exercise 2.1 Come up with a dynamical system representing a falling rock. Specify each component of the tuple $\mathcal{D} = (\mathcal{U}, \mathcal{Y}, \Sigma, \varphi, r)$, as well as the time set $\mathcal{T}$. Explain why the time, restriction, semigroup, and identity axioms hold for this system.

Exercise 2.2 Come up with a dynamical system representing an airplane. Specify each component of the tuple $\mathcal{D} = (\mathcal{U}, \mathcal{Y}, \Sigma, \varphi, r)$, as well as the time set $\mathcal{T}$. Explain why the time, restriction, semigroup, and identity axioms hold for this system.

Exercise 2.3 Come up with a dynamical system representing an computer. Specify each component of the tuple $\mathcal{D} = (\mathcal{U}, \mathcal{Y}, \Sigma, \varphi, r)$, as well as the time set $\mathcal{T}$. Explain why the time, restriction, semigroup, and identity axioms hold for this system.

These three examples - of a falling rock, an airplane, and a computer - illustrate the flexibility of the dynamical system definition we posed above. Each system is vastly different, yet fits into the same framework under minimal assumptions.

However, this flexibility comes at a price. Whenever one makes a highly abstract, general definition such as Definition 2.1, there is a fundamental tradeoff that one makes. Almost always, generality comes at the expense of *practicality* - the more general the definition, the less practical it is, and the harder it is to come to interesting conclusions. In order to state more interesting results about the behavior of dynamical systems, we'll need to consider dynamical systems with more structure than the basic scaffolding offered by Definition 2.1.

First, we outline two simple classes of dynamical system, based on the examples of an airplane and a computer. In order to properly describe the behavior of an airplane, one needs to use a time set $\mathcal{T} = \mathbb{R}$. On the other hand, for a computer—which makes decisions at discrete instants—one might use the time set $\mathcal{T} = \mathbb{Z}$. We distinguish between systems on these two important time sets as follows.

Definition 2.2 (Continuous-Time System) A dynamical system $\mathcal{D}$ is said to be a continuous-time system if its time set is $\mathcal{T} = \mathbb{R}$.

Definition 2.3 (Discrete-Time System) A dynamical system $\mathcal{D}$ is said to be a discrete-time system if its time set is $\mathcal{T} = \mathbb{Z}$.

As a general rule of thumb, if the state of a system varies as time passes in seconds (with no fixed jumps or increments in time), the system will be continuous-time. On the other hand, if the state of a system jumps at fixed, discrete increments, the system will be discrete-time[1].

These two classes of system provide some nice, additional structure to Definition 2.1. What other structure might be interesting to add? Let's think back to the example of a falling rock, and see if anything else jumps out at us that isn't explicitly covered by Definition 2.1.

Suppose the state of the rock is its position and velocity in space and that its output is its position. As an input, let's take a force acting on the rock. This enables us to treat the action of "dropping" the rock as an input signal. To track the trajectory of the rock after

[1] Note that the choice of $\mathcal{T} = \mathbb{Z}$ for discrete-time is somewhat arbitrary - one could reasonably replace $\mathbb{Z}$ with another countable subset of $\mathbb{R}$ that satisfies the time axiom.

we drop it, we could check the readout map. This tells us that, if we drop the rock at time t_0 from initial state x_0, the rock position at time t is,

$$\text{rock position at time } t = r(t, \varphi(t, t_0, x_0, u(\cdot)), u(t)). \tag{2.9}$$

What if, instead of dropping the rock at time t_0, we sat around for ten minutes and dropped the rock at time $t_0' = t_0 + 10$? Would we expect the rock to fall in the same way? If our rock is friends with Isaac Newton, the answer is—of course! To determine the trajectory of the rock, we shouldn't need to know the number of seconds since the beginning of time at which we drop it—what matters is how much time has *passed* since we dropped it. This behavior—of time *passed* being the relevant quantity of time—is prevalent in a number of systems. A dynamical system possessing this property is said to be *time-invariant.*

Before we can properly define time invariance, we must make a few auxiliary definitions. First, we make a definition that will reduce our notational overhead. We note that in the example of the falling rock, we composed the readout map with the state transition map in order to get the output at time t, given an initial time, initial state, and input signal. Since it's quite cumbersome to rewrite this composition every time we're interested in these objects, we define the following map.

Definition 2.4 (Input/Output Map) Given a dynamical system $\mathcal{D}$, the input/output map (I/O map) is the map $\rho : \mathbf{T} \times \Sigma \times \mathcal{U} \to Y$ (for $\mathbf{T} = \{(t_1, t_0) \in \mathcal{T} \times \mathcal{T} : t_1 \geq t_0\}$) defined as the composition,

$$\rho(t_1, t_0, x_0, u(\cdot)) = r(t_1, \varphi(t_1, t_0, x_0, u(\cdot)), u(t_1)), \tag{2.10}$$

of the readout and the state transition maps.

Remark 2.6 An I/O map takes in a pair of times, an initial state, and an input signal and returns an output *value*, not an output signal! This is because we want the I/O map to convey information about the output at a particular time, rather than at all times.

With this definition made, we turn our attention back to the problem of defining time-invariance. First, we focus on how the components of a dynamical system change over time. We define a *delay-invariant* set of signals.

Definition 2.5 (Delay-Invariant Set) Consider a set of signals, $\mathcal{U} \subseteq \{u : \mathcal{T} \to U\}$, where $\mathcal{T} \subseteq \mathbb{R}$ is a time set and U is an arbitrary set. If, for all $\tau \in \mathcal{T}$ and all $u(\cdot) \in \mathcal{U}$, the signal

$$\hat{u} : \mathcal{T} \to U, \ \hat{u}(t) = u(t - \tau), \tag{2.11}$$

also belongs to $\mathcal{U}$, then $\mathcal{U}$ is said to be a *delay-invariant set* with respect to $\mathcal{T}$.

Remark 2.7 When the time set $\mathcal{T}$ is clear from context, one refers to a delay-invariant set with respect to $\mathcal{T}$ simply as a "delay-invariant set." The "with respect to $\mathcal{T}$" can be dropped.

Remark 2.8 Delay-invariant sets are so-called since they are defined by delaying signals by a time τ. It's important to note that if $\tau < 0$, a "delay" will actually shift a signal *forward* in time rather than shifting it backward. In this context, the name "delay" is therefore not entirely consistent with our intuitive understanding of the word.

Remark 2.9 In this definition, we implicitly make use of the time axiom. Without the guarantee that $t_1 - t_2 \in \mathcal{T} \ \forall t_1, t_2 \in \mathcal{T}$, we would not know $\hat{u}(t) = u(t - \tau)$ was a valid signal.

Thus, a set is delay-invariant if any signal in $\mathcal{U}$ can be delayed by any time τ and remain in $\mathcal{U}$. Equipped with this definition, we define a *delay map* on a delay-invariant set of signals.

Definition 2.6 (Delay Map) Consider a delay-invariant set of signals, $\mathcal{U}$. For $\tau \in \mathcal{T}$, the map $T_\tau : \mathcal{U} \to \mathcal{U}$, defined $(T_\tau(u))(t) = u(t-\tau)\ \forall t \in \mathcal{T}$, is called the delay map of time τ.

Based on this definition, a shift map T_τ simply *delays* any input signal by a fixed time τ. Notice how the definition of a delay-invariant set ensures the delay map is well-defined—since we don't have to worry about a delayed signal leaving the set $\mathcal{U}$, we define the delay map to be a map from $\mathcal{U} \to \mathcal{U}$.

Definition 2.7 (Time-Invariant System) A causal I/O dynamical system $\mathcal{D}$ is said to be time invariant if, for all $\tau \in \mathcal{T}$, the following conditions are satisfied:

1. Delay-invariant input space: $\mathcal{U}$ is a delay-invariant set.
2. Delay-invariant output space: $\mathcal{Y}$ is a delay-invariant set.
3. Delay-invariant transition map: For all $t_0, t_1, \tau \in \mathcal{T}$ with $t_0 \leq t_1$ and all $x_0 \in \Sigma$, $u(\cdot) \in \mathcal{U}$,

$$\rho(t_1, t_0, x_0, u(\cdot)) = \rho(t_1+\tau, t_0+\tau, x_0, T_\tau(u(\cdot))), \tag{2.12}$$

where $T_\tau : \mathcal{U} \to \mathcal{U}$ is the delay map of time τ on $\mathcal{U}$ and ρ is the I/O map of $\mathcal{D}$.

Item (3) of this definition—delay-invariant transition map—is by far the most important component of the definition. It states that the output of the system depends on the amount of time that has *passed*, rather than on the explicit start and end times. In particular, if we delay the inputs to the system by time τ, we will get the same output at time $t+\tau$ as the undelayed system at time t.

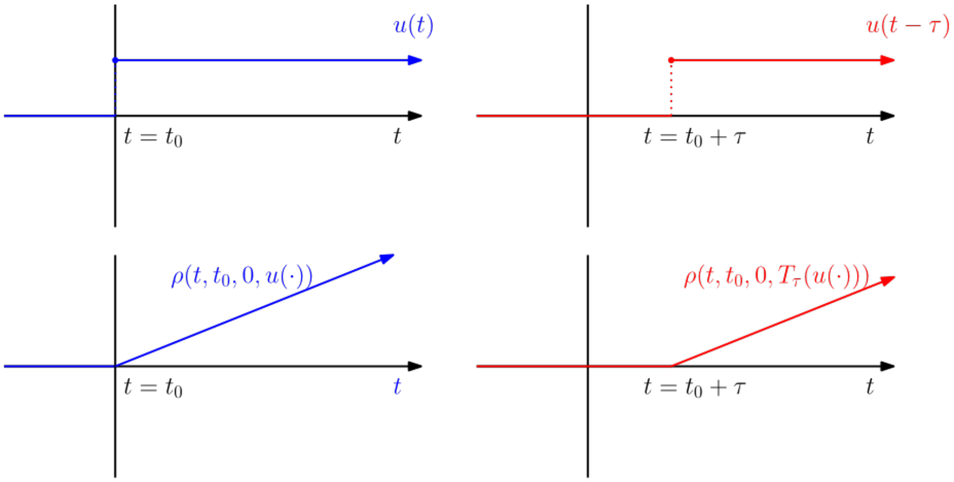

Fig. 2.3 An example of a time-invariant response. On the left-hand side, we apply an input which jumps up at time $t = t_0$. The system responds by increasing along a ramp at time $t = t_0$. If we *delay* the input by τ, the appearance of the ramp also delays by τ. Thus, we observe that, for this input, the system respects the equality $\rho(t+\tau, t_0+\tau, T_\tau(u(\cdot))) = \rho(t, t, t_0, u(\cdot))$.

What other structure can we add to a causal I/O dynamical system? Thus far, we've only placed structure on the *time* component of the I/O map, and haven't considered any algebraic or analytic conditions.

Although algebraically and analytically unstructured maps lend themselves well to generality, one cannot say the same for practicality. Without placing further algebraic or analytic constraints on the I/O map, we'll find it hard to perform any meaningful system analysis. For the class of *linear* I/O systems, however, a wide array of concepts become mathematically and computationally tractable. This is the class of systems we will focus on in this course.

Definition 2.8 (Linear I/O System) Consider an I/O system $\mathcal{D} = (\mathcal{U}, \mathcal{Y}, \Sigma, \varphi, r)$ with I/O map ρ. $\mathcal{D}$ is said to be a linear I/O system if the following conditions are satisfied:

1. Linear Spaces: $\mathcal{U}, \mathcal{Y}$, and Σ are vector spaces over a common field, $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$.
2. Linear I/O Map: For each fixed pair $t_0, t_1 \in \mathcal{T}$ with $t_0 \leq t_1$, the I/O map is linear in $\Sigma \times \mathcal{U}$. That is, for all $x_0, \hat{x}_0 \in \Sigma$, $u(\cdot), \hat{u}(\cdot) \in \mathcal{U}$, and $\alpha, \beta \in \mathbb{K}$,

$$\rho(t_1, t_0, \alpha x_0 + \beta \hat{x}_0, \alpha u(\cdot) + \beta \hat{u}(\cdot)) = \alpha \rho(t_1, t_0, x_0, u(\cdot)) + \beta \rho(t_1, t_0, \hat{x}_0, \hat{u}(\cdot)). \tag{2.13}$$

Remark 2.10 If the spaces $\mathcal{U}, \mathcal{Y}, \Sigma$ of $\mathcal{D}$ are all over a field $\mathbb{K}$, one says that $\mathcal{D}$ itself is a system over a field $\mathbb{K}$.

Remark 2.11 We'll refer to a linear I/O system simply as a "linear system" or a"linear dynamical system" where context allows.

Let's discuss the different components of the definition. The first condition, *linear spaces*, states that the input and output *signal* spaces are vector spaces, as is the state space Σ. This means that any linear combination of input signals, $\alpha u(\cdot) + \beta \hat{u}(\cdot)$, remains in the input space. Likewise, linear combinations of output signals and states remain in the output and state spaces, respectively.

The second condition, *linear I/O map*, states that the output of a linear I/O system must be linear in its initial condition and input. That is, if we scale the initial condition and output by the same value, the output should scale by that value as well. Additionally, if we add two sets of initial conditions and inputs, the corresponding output should be the sum of the individual outputs.

Let's get a basic feel for what this linear structure enables. In the following proposition, we state a few simple consequences of Definition 2.8.

Proposition 2.1 (Output Response of Linear I/O Systems) *Any linear I/O system $\mathcal{D}$ over a field $\mathbb{K}$ satisfies the following:*

1. *Zero Input Response: For all $x_0, \hat{x}_0 \in \Sigma$, $t_0, t_1 \in \mathcal{T}$ with $t_0 \leq t_1$, and $\alpha, \beta \in \mathbb{K}$,*

$$\rho(t_1, t_0, \alpha x_0 + \beta \hat{x}_0, 0) = \alpha \rho(t_1, t_0, x_0, 0) + \beta \rho(t_1, t_0, \hat{x}_0, 0). \tag{2.14}$$

2. *Zero State Response: For all $u(\cdot), \hat{u}(\cdot) \in \mathcal{U}$, $t_0, t_1 \in \mathcal{T}$ with $t_0 \leq t_1$, and $\alpha, \beta \in \mathbb{K}$,*

$$\rho(t_1, t_0, 0, \alpha u(\cdot) + \beta \hat{u}(\cdot)) = \alpha \rho(t_1, t_0, 0, u(\cdot)) + \beta \rho(t_1, t_0, 0, \hat{u}(\cdot)). \tag{2.15}$$

3. *Zero Input/Zero State Decomposition: For all $x_0 \in \Sigma$, $u(\cdot) \in \mathcal{U}$, and $t_0, t_1 \in \mathcal{T}$ with $t_0 \leq t_1$,*

$$\rho(t_1, t_0, x_0, u(\cdot)) = \rho(t_1, t_0, x_0, 0) + \rho(t_1, t_0, 0, u(\cdot)). \tag{2.16}$$

Here, $\rho(t_1, t_0, x_0, 0)$ is called the zero-input response and $\rho(t_1, t_0, 0, u(\cdot))$ the zero-state response.

The last item of Proposition 2.1 tells us that, in order to understand the response of a linear I/O system to *any* input, all we need is the zero-input response and the zero-state response—there is an exact decomposition of the total response into the zero-input and zero-state components.

Exercise 2.4 Prove Proposition 2.1.

Using our earlier definition of time-invariance, one may determine a further classification of linear dynamical systems.

Definition 2.9 (Linear Time-Invariant/Varying System) An I/O system $\mathcal{D}$ is said to be linear, time-invariant (LTI) if it is a linear I/O system and it is time-invariant. If a linear I/O system is not necessarily linear, time-invariant, it is said to be linear, time-varying (LTV).

2.1.2 State Space Representations of Linear Systems

Thus far, we've only dealt with *abstract* dynamical systems, in which the evolution of the system is described by an arbitrary state transition map. Is this the most convenient way of describing a dynamical system? In practice, dynamical systems are typically not specified via their state transition map. Instead, one often specifies a set of equations (such as a differential equation or a recurrence relation) from which a state transition map can be determined. If we wish to describe a Newtonian physical system, for instance, we might start by writing down Newton's second law, $F = m\ddot{x}$, and deriving *differential equations* of motion. How do systems with dynamics of this form correspond to the abstract dynamical systems we discussed above?

In order to establish this connection, we make the important distinction between a *representation* of a dynamical system and the dynamical system itself. When we write down a differential equation such as $F = m\ddot{x}$, we define a differential equation *representation* of an abstract dynamical system. More generally, we say that a *system representation* is a description of a dynamical system using some mathematical framework (such as an ordinary differential equation, partial differential equation, recurrence relation, etc.) that fully determines the dynamical system.

Definition 2.10 ((Informal) I/O System Representation) An I/O system representation is a collection of mathematical data that uniquely determines a causal I/O dynamical system.

Let's return to the example of a physical system described by $F = m\ddot{x}$ to illustrate what we mean by this. Let's take F to be the input to the dynamical system, $(x, \dot{x})$ to be the state, and x to be the output. The pair,

$$\ddot{x} = \frac{1}{m}F, \quad r(x, \dot{x}, F) = x, \tag{2.17}$$

of an ordinary differential equation $\ddot{x} = \frac{1}{m}F$ and an readout function $r(x, \dot{x}, F) = x$, together with sets of admissible inputs and outputs, constitute a representation of the abstract dynamical system. The solutions of the differential equation uniquely determine the state transition map of the dynamical system while the equation $r(x, \dot{x}, F) = x$ determines the readout map. Thus, the physical system is *represented* by a differential equation, a readout map, and input and output spaces.

This simple example leads us to ask a few important questions regarding representations of dynamical systems. What are common representations of dynamical systems? Linear dynamical systems? Linear, time-invariant dynamical systems?

We'll first answer these questions for the continuous-time case, in which $\mathcal{T} = \mathbb{R}$. In order to present a well-posed definition for system representations of continuous-time linear systems, we first need to define a special class of signals: *piecewise continuous signals*. In the next section, we'll find that piecewise continuous signals are the "right" class of input signal for continuous-time linear system representations.

Definition 2.11 (Piecewise Continuity) Let V be a normed vector space and $I \subseteq \mathbb{R}$ a (possibly infinite) interval. A mapping $u : I \to V$ is said to be piecewise continuous on I if there exists a set $D \subseteq I$, called the discontinuity set, for which the following hold:

1. Continuity outside D: u is continuous on $I \setminus D$.
2. Left and right limits: for all $\tau \in D$, the left and right limits $\lim_{t\to\tau^-} u(t)$ and $\lim_{t\to\tau^+} u(t)$ exist and are finite.
3. Finite intersections: for all $t_0, t_1 \in \mathbb{R}$ with $t_0 \leq t_1$, the set $D \cap [t_0, t_1]$ contains a finite number of points.

The set of all piecewise continuous mappings from I into V is denoted $PC(I, V)$.

Remark 2.12 The above is sometimes referred to as "piecewise continuity with one-sided limits." Here, we incorporate the one-sided limits into the definition of piecewise continuity.

Example 2.1 (Unit Step Function) The unit step function, $\mathbb{1} : \mathbb{R} \to \mathbb{R}$, defined

$$\mathbb{1}(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0, \end{cases} \tag{2.18}$$

is a piecewise continuous function in $PC(\mathbb{R}, \mathbb{R})$, with discontinuity set $D = \{0\}$.

Exercise 2.5 Verify that any continuous mapping $f : I \to V$ is piecewise continuous.

Importantly, the set of piecewise continuous functions has a natural vector space structure.

Proposition 2.2 ($PC(I, V)$ is a Vector Space) *Let $I \subseteq \mathbb{R}$ a nonempty interval and V a normed vector space over $\mathbb{K}$. When equipped with the operations $+$ of function addition and $(\cdot)$ of scalar multiplication of functions, $PC(I, V)$ forms a vector space over $\mathbb{K}$.*

Proof See Problem 2.2. □

With these definitions in our toolbelt, we formulate a precise, well-posed definition for a continuous-time LTV system representation.

Definition 2.12 (Continuous-Time LTV System Representation) A continuous-time LTV system representation consists of the following data:

1. Input, output, and state spaces: an input space $\mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m)$, output space $\mathcal{Y} = PC(\mathbb{R}, \mathbb{R}^p)$, and state space $\Sigma = \mathbb{R}^n$.

2. Matrix functions: matrix functions $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, and $D(\cdot)$ satisfying,

$$A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times n}),\ B(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times m}), \tag{2.19}$$
$$C(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{p\times n}),\ D(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{p\times m}). \tag{2.20}$$

3. State & output equations: a differential equation and an algebraic equation,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \text{ (state equation)} \tag{2.21}$$
$$y(t) = C(t)x(t) + D(t)u(t) \text{ (output equation)}, \tag{2.22}$$

where $x(t) \in \mathbb{R}^n$, $u(\cdot) \in \mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m)$, and $y(\cdot) \in \mathcal{Y} = PC(\mathbb{R}, \mathbb{R}^p)$.

We refer to the system representation by the tuple $(A(\cdot), B(\cdot), C(\cdot), D(\cdot))$. Such a representation is said to be a continuous-time, *state-space* system representation. The vector $x(t)$ is referred to as the *state vector*, $u(t)$ as the *input vector*, and $y(t)$ as the *output vector*.

Remark 2.13 Since the input space of a continuous-time LTV system representation is $\mathcal{U} = PC(\mathbb{R}, \mathbb{R}^n)$, the input-value space is $U = \mathbb{R}^m$. Likewise, the output-value space is $Y = \mathbb{R}^p$.

This definition has a lot of moving parts, so let's take a moment to summarize the key points. A continuous time, linear time-varying system is defined by a tuple $(A(\cdot), B(\cdot), C(\cdot), D(\cdot))$ of *matrix-valued* functions, each of which is piecewise continuous. The time set of the system is $\mathbb{R}$, which makes it a continuous-time system. The state space of such a system is $\mathbb{R}^n$, while the input and output spaces are $PC(\mathbb{R}, \mathbb{R}^m)$ and $PC(\mathbb{R}, \mathbb{R}^p)$. Since the system is described by a pair of a state equation (the differential equation $\dot{x} = A(t)x + B(t)u$) and an output equation (the algebraic equation $y = C(t)x + D(t)u$), such a representation is referred to as a *state-space* representation.

Why the emphasis on piecewise continuity? As we'll see in the next section, the piecewise continuity assumption is *essential* for the system representation to determine a unique dynamical system. Without this assumption, we aren't guaranteed to have unique solutions to the differential equation $\dot{x} = A(t)x + B(t)u$, which would cause us trouble in defining a state transition map.

Now that we've defined a linear, time-varying system representation, we have an enormous open question on our hands:

Does Definition 2.12 determine a formal linear I/O dynamical system
in the sense of Definition 2.8?

In order to answer this question, one must perform a nontrivial study of the existence and uniqueness of solutions to differential equations. Nevertheless, we will find in the next section that, after performing this study, Definition 2.12 does indeed yield a valid system representation. More on this later!

Let's write down a few more important classes of system representations. Now that we've defined a continuous time, linear time-*varying* system representation, it's only natural to define a continuous time, linear time-*invariant* system representation.

Definition 2.13 (Continuous-Time LTI System Representation) A continuous-time LTI system representation consists of the following data:

1. Input, output, and state spaces: an input space $\mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m)$, output space $\mathcal{Y} = PC(\mathbb{R}, \mathbb{R}^p)$, and state space $\Sigma = \mathbb{R}^n$.

2. Matrices: fixed matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$.
3. State & output equations: a differential equation and an algebraic equation,

$$\dot{x}(t) = Ax(t) + Bu(t) \text{ (state equation)} \tag{2.23}$$
$$y(t) = Cx(t) + Du(t) \text{ (output equation),} \tag{2.24}$$

   where $x(t) \in \mathbb{R}^n$, $u(\cdot) \in \mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m)$, and $y(\cdot) \in \mathcal{Y} = PC(\mathbb{R}, \mathbb{R}^p)$.

We refer to the system representation by the tuple (A, B, C, D).

Thus, in order to define a continuous time, linear time *invariant* system, we simply take the definition of a continuous time, linear time varying system and remove all dependence on time from the matrix functions $(A(\cdot), B(\cdot), C(\cdot), D(\cdot))$. Once again, we face a question regarding how this representation relates to the LTI systems we defined above.

Does Definition 2.13 determine a formal LTI I/O dynamical system
in the sense of Definition 2.9?

Fortunately, we'll find that the answer is yes! As with the above, we'll wait until the next section to prove this.

Now, we define discrete-time analogues of Definitions 2.12 and 2.13. Since the time set of a discrete-time system is $\mathbb{Z}$, we can drop all of the piecewise continuity assumptions on matrix functions and signals when defining discrete-time system representations.

Definition 2.14 (Discrete-Time LTV System Representation) A discrete-time LTV system representation consists of the following data:

1. Input, output and state spaces: an input space $\mathcal{U} = \{u : \mathbb{Z} \to \mathbb{R}^m\}$ and output space $\mathcal{Y} = \{y : \mathbb{Z} \to \mathbb{R}^p\}$ of all functions from $\mathbb{Z}$ to $\mathbb{R}^m$ and $\mathbb{R}^p$, and state space $\Sigma = \mathbb{R}^n$.
2. Matrix functions: matrix-valued functions $A[\cdot], B[\cdot], C[\cdot], D[\cdot]$,

$$A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n},\ B[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times m} \tag{2.25}$$
$$C[\cdot] : \mathbb{Z} \to \mathbb{R}^{p\times n},\ D[\cdot] : \mathbb{Z} \to \mathbb{R}^{p\times m}. \tag{2.26}$$

3. State & output equations: a recurrence relation and an algebraic equation,

$$x[k+1] = A[k]x[k] + B[k]u[k] \text{ (state equation)} \tag{2.27}$$
$$y[k] = C[k]x[k] + D[k]u[k] \text{ (output equation),} \tag{2.28}$$

   where $x[k] \in \mathbb{R}^n$, $u[\cdot] \in \mathcal{U} = \{u : \mathbb{Z} \to \mathbb{R}^m\}$, and $y[\cdot] \in \mathcal{Y} = \{y : \mathbb{Z} \to \mathbb{R}^p\}$.

We refer to the system representation by the tuple $(A[\cdot], B[\cdot], C[\cdot], D[\cdot])$. Such a system is said to be a discrete time, *state-space* system representation. The vector $x[k]$ is referred to as the *state vector*, $u[k]$ as the *input vector*, and $y[k]$ as the *output vector*.

Remark 2.14 Note that there are several popular ways of writing the state and output equations of a discrete-time system. Above, we've used square brackets to describe the time, $k \in \mathbb{Z}$. Other popular notation includes,

$$x_{k+1} = A_k x_k + B_k u_k \qquad x(k+1) = A(k)x(k) + B(k)u(k) \tag{2.29}$$
$$y_k = C_k x_k + D_k u_k \qquad y(k) = C(k)x(k) + D(k)u(k). \tag{2.30}$$

Remark 2.15 In Definition 2.14, we've defined inputs, outputs, and matrix-valued functions to be functions from $\mathbb{Z}$ into each of their respective spaces. This means that we can view the inputs, outputs, and matrix functions as *sequences*.

Definition 2.15 (Discrete-Time LTI System Representation) A discrete-time LTI system representation consists of the following data:

1. Input, output, and state spaces: an input space $\mathcal{U} = \{u : \mathbb{Z} \to \mathbb{R}^m\}$, output space $\mathcal{Y} = \{y : \mathbb{Z} \to \mathbb{R}^p\}$, and state space $\Sigma = \mathbb{R}^n$.
2. Matrices: fixed matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$.
3. State & output equations: a recurrence relation and an algebraic equation,

$$x[k+1] = Ax[k] + Bu[k] \text{ (state equation)} \tag{2.31}$$

$$y[k] = Cx[k] + Du[k] \text{ (output equation)}, \tag{2.32}$$

   where $x[k] \in \mathbb{R}^n$, $u[\cdot] \in \mathcal{U} = \{u : \mathbb{Z} \to \mathbb{R}^m\}$, and $y[\cdot] \in \mathcal{Y} = \{y : \mathbb{Z} \to \mathbb{R}^p\}$.

We refer to the system representation by the tuple (A, B, C, D).

The fact that we don't have to worry about regularity conditions on our signals (as we did with piecewise-continuity in continuous-time) hints that the analysis of discrete-time systems might be easier than the analogous analysis of continuous-time systems. This is in fact the case in a number of scenarios.

To wrap this section up, we define SISO and MIMO systems. Frequently, we'll distinguish between systems that only have a single input and output (which are generally easier to analyze) and systems that have multiple inputs and outputs.

Definition 2.16 (SISO/MIMO System Representations) Consider a continuous or discrete-time system representation in which

$$\mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m) \text{ or } \mathcal{U} = \{u : \mathbb{Z} \to \mathbb{R}^m\} \tag{2.33}$$

$$\mathcal{Y} = PC(\mathbb{R}, \mathbb{R}^p) \text{ or } \mathcal{Y} = \{y : \mathbb{Z} \to \mathbb{R}^p\}. \tag{2.34}$$

If $m = p = 1$, the system is said to be single-input, single-output (SISO). If $m, p \geq 1$, the system is said to be multi-input, multi-output (MIMO).

Remark 2.16 Note that we don't take the definition of MIMO to be strictly greater than 1—this way, we can consider MIMO to be a straightforward generalization of SISO.

Based on this definition, SISO systems appear to be scalar from the input-output perspective: we put in a scalar as an input and get another scalar as an output. It's important to note that the SISO/MIMO distinction has *nothing* to do with the dimension of the state space! One can have a SISO system with an arbitrarily high-dimension state space, provided the input and output-value spaces are both one-dimensional.

2.1.3 Further Reading

This section was mainly influenced by [29], [22], and [6]. Our development of abstract dynamical systems most closely follows that of Chapter 5 of [6]. For a more in-depth look at abstract dynamical systems, we refer the reader to Chapter 2 of [29]. The example of a shower control system, used all the way at the start of the section, is from [24].

2.1.4 Problems

Problem 2.1 (Causal & Noncausal Maps [2]) As we mentioned in the section above, one can represent the input/output relationship of a system for a *fixed* initial time and state directly with a function $H : \mathcal{T} \times \mathcal{U} \to \mathcal{Y}$. That is, one has $y(t) = H(t, u(\cdot))$ for any time t and admissible input $u(\cdot)$. In this problem, we'll determine definitions for causality, linearity, and time-invariance of an arbitrary map $H : \mathcal{T} \times \mathcal{U} \to \mathcal{Y}$.

1. Given an arbitrary map $H : \mathcal{T} \times \mathcal{U} \to \mathcal{Y}$, formulate a definition of *time-invariance* for H. Formulate a definition of *causality*. Formulate a definition of *linearity. Hint: for causality, think about the restriction of a signal to a certain time interval.*
2. Let's put our definitions to the test. In each of the following cases, determine whether the system is causal/time-invariant/linear. Use your best judgment to identify the input and output spaces in each case.

   a. Consider a discrete-time system with I/O description $y[k] = c_1 u[k+1] + c_2$, where $c_1, c_2 \in \mathbb{R}$. Is this system causal? Is it time-invariant? Is it linear?
   b. Consider a continuous-time system with I/O description $y(t) = u(t - \tau)$, where $\tau \in \mathbb{R}$ is fixed and positive. Is this system causal? Is it time invariant? Is it linear?
   c. Consider a continuous-time system with I/O description,

   $$y(t) = \begin{cases} u(t) & t \leq \tau \\ 0 & t > \tau. \end{cases} \tag{2.35}$$

   Is this system causal? Is it time-invariant? Is it linear?
   d. Consider a continuous-time system with I/O description,

   $$y(t) = \min\{u_1(t), u_2(t)\}, \tag{2.36}$$

   where $u(t) = [u_1(t); u_2(t)]^\top$ is the system input. Is this system causal? Is it time-invariant? Is it linear?

Problem 2.2 (Properties of Piecewise-Continuous Functions) In the section above, we introduced the class of piecewise-continuous functions. In this problem, we'll prove some basic properties of this function class.

1. Show that $PC(\mathbb{R}, \mathbb{R}^n)$ forms a vector space over $\mathbb{R}$ under the operations of function addition and scalar multiplication.
2. Let $I, K \subseteq \mathbb{R}$ be compact intervals. Show that any $f \in PC(I, \mathbb{R})$ must be bounded above on $I \cap K$,

   $$\sup_{t \in I \cap K} f(t) < \infty. \tag{2.37}$$

3. Let $I \subseteq \mathbb{R}$ be a compact interval and $\|\cdot\|$ be an arbitrary norm on $\mathbb{R}^n$. Show that the supremum norm,

   $$\|f\|_\infty = \sup_{t \in I} \|f(t)\|, \tag{2.38}$$

   is finite for all $f \in PC(I, \mathbb{R}^n)$. Then, prove that $\|\cdot\|_\infty$ makes $PC(I, \mathbb{R}^n)$ into a normed vector space.

4. Is $PC(I, \mathbb{R}^n)$ a Banach space with respect to the supremum norm $\|\cdot\|_\infty$, $\|f\|_\infty = \sup_{t\in\mathbb{R}} \|f(t)\|$? Provide a proof or a counterexample.

2.2 Solutions of Linear, Time-Varying Systems

Now that we've introduced a set of state space representations of linear systems, we must show that these representations *are* in fact representations in the formal sense. Recall that in the previous section, we posed two questions:

Does an LTV system representation determine a linear I/O system?
Does an LTI system representation determine an LTI I/O system?

Further, we promised that in this section, we would provide a *precise* answer to both of these queries. Now that we're here, we need to make good on this promise! In this section, we establish answers to these questions by studying the existence, uniqueness, and structure of solutions to linear ordinary differential equations and recurrence relations. We'll then apply the results of this study to answer the two questions above.

In order to answer these questions, we'll split up into the continuous and discrete-time cases. In order to study continuous-time linear systems, we must study linear ordinary differential equations, while to study discrete-time linear systems, we must study linear recurrence relations. Along the way, we'll draw connections between the techniques used to study the two. Let's begin!

2.2.1 Solutions of Continuous-Time Linear Systems

We'll begin by laying out a brief plan of attack for our study of continuous-time linear system representations. Recall from the previous section that a continuous-time LTV system representation is specified by a tuple $(A(\cdot), B(\cdot), C(\cdot), D(\cdot))$ of piecewise continuous matrix-valued functions. These functions define two equations,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \text{ (state equation)} \tag{2.39}$$

$$y(t) = C(t)x(t) + D(t)u(t) \text{ (output equation)}, \tag{2.40}$$

which govern how the state $x(t)$ and the output $y(t)$ change over time as input signals $u : \mathbb{R} \to \mathbb{R}^m$ are applied to the system.

In order to determine if the LTV system representation yields a valid linear input/output dynamical system in the sense of Definition 2.8, we must tick a couple of boxes. To verify that the LTV system representation yields a linear I/O system, we must compute the I/O map, ρ, associated to the representation, and verify that it satisfies the linearity conditions proposed in the previous section. In order to compute ρ, however, we require the state transition map, φ, of the representation.

Thus, we begin by studying the state transition map. The state transition map associated to the LTV system representation maps from times $t_0, t_1 \in \mathbb{R}$ with $t_0 \leq t_1$, initial state $x_0 \in \mathbb{R}^n$, and input signal $u \in PC(\mathbb{R}, \mathbb{R}^m)$ to the solution of the differential equation,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{2.41}$$

at time t_1, with initial condition $x(t_0) = x_0$. Therefore, in order to have a state transition map $\varphi(t_1, t_0, x_0, u(\cdot))$ that is well-defined on the input space $\mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m)$, there are a couple of things we need:

1. Existence: for all inputs $u \in PC(\mathbb{R}, \mathbb{R}^m)$, initial conditions $x_0 \in \mathbb{R}^n$, and times $t_0 \in \mathbb{R}$, a solution to the initial value problem,
$$\dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(t_0) = x_0, \tag{2.42}$$
must exist. This ensures that we will always be able to compute φ on the time set, state space, and input space of the representation.
2. Uniqueness: for all inputs $u \in PC(\mathbb{R}, \mathbb{R}^m)$, initial conditions $x_0 \in \mathbb{R}^n$, and times $t_0 \in \mathbb{R}$, the solution to the initial value problem,
$$\dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(t_0) = x_0, \tag{2.43}$$
must be *unique*. If we want the state transition map to be well-defined, we can't have two or more solutions to the initial value problem!

Thus, in order to solve the problem of verifying the LTV system representation yields a linear I/O system, we must first establish the existence and uniqueness of solutions to the initial value problem $\dot{x} = A(t)x(t) + B(t)u(t),\ x(t_0) = x_0$. We'll break this down into the following multi-step process:

1. Define Solutions: first, we'll formulate a precise definition for a solution to a time-varying initial value problem with piecewise continuous data.
2. Matrix IVP: next, we'll argue that it will be insightful to study solutions to a simpler yet more fundamental initial value problem, the *matrix* initial value problem
$$\dot{X}(t) = A(t)X(t),\ X(t_0) = I. \tag{2.44}$$
We will prove that this problem has a unique solution and will study its basic structure.
3. State Transition Matrix: after showing that a unique solution to the IVP $\dot{X}(t) = A(t)X(t),\ X(t_0) = I$ exists, we'll *abstract away* details of the solution into a special operator called the state transition matrix. Then, we'll establish a few important properties of the state transition matrix.
4. LTV-IVP: finally, we'll show that we can use the state transition matrix to study solutions to the general linear time-varying, initial value problem. Here, we'll complete the problem of proving existence and uniqueness of solutions.

2.2.1.1 Defining Solutions to IVPs

Let's tackle the first step of the process we outlined above. What does it mean to solve a time-varying initial value problem with piecewise continuous data? Although it might *seem* like all we need is to find a differentiable function which satisfies $\dot{x}(t) = A(t)x(t) + B(t)u(t)$ for all $t \in \mathbb{R}$ and $x(t_0) = x_0$, the reality is somewhat more complex! To illustrate what goes wrong with this "naive" definition of a solution, consider the case of a simple scalar, time-varying initial value problem,
$$\dot{x}(t) = b(t),\ x(0) = 0. \tag{2.45}$$

Suppose $b \in PC(\mathbb{R}, \mathbb{R})$ is the *step function*, the function which is identically zero before $t = 0$ and identically one at and after $t = 0$,

$$b(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0. \end{cases} \tag{2.46}$$

Using our basic knowledge of ODEs, we know that the solution to this initial value problem *should* be the ramp function,

$$x(t) = \begin{cases} t & t \geq 0 \\ 0 & t < 0. \end{cases} \tag{2.47}$$

However, this function is clearly *not* differentiable at the point $t = 0$—right where b makes its jump, we find that the proposed solution of the initial value problem has a sharp "corner." As a result of this, we find that requiring a solution to this IVP to be a *differentiable* function $x : \mathbb{R} \to \mathbb{R}$ satisfying $\dot{x}(t) = b(t)\ \forall t$, $x(0) = 0$, is *too strong.*

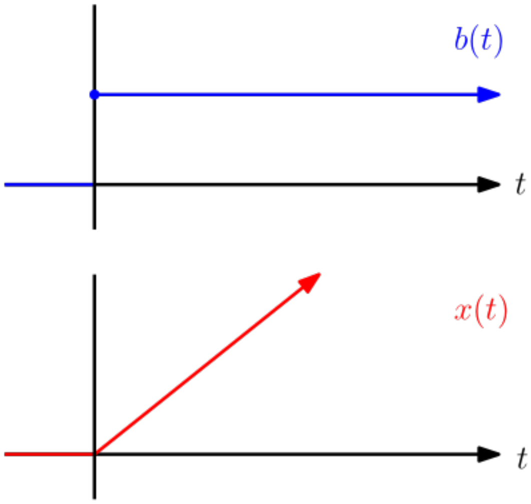

Fig. 2.4 The solution to the initial value problem $\dot{x}(t) = b(t)$, with b the step function, *should* be the ramp function. However, the ramp function is *not* differentiable at $t = 0$! Thus, our definition for a solution an IVP must account for points of non-differentiability.

This example is a particular instance of a general fact from analysis: if a function has a jump discontinuity, it *cannot* be the derivative of another function (see Problem 2.4 for the formal details of this argument). We conclude that, in order to make a *well-posed* definition for a solution to an initial value problem with piecewise continuous data, we must explicitly account for the points of discontinuity. This leads us to the following, formal definition of a solution.

Definition 2.17 (Solution to LTV-IVP) Consider the piecewise-continuous maps $A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n \times n})$, $B(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n \times m})$, and $u(\cdot) \in PC(\mathbb{R}, \mathbb{R}^m)$, which have a shared discontinuity set $D \subseteq \mathbb{R}$. For $x_0 \in \mathbb{R}^n$ and $t_0 \in \mathbb{R}$, a solution to the initial value problem,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(t_0) = x_0, \tag{2.48}$$

is a continuous map $x : \mathbb{R} \to \mathbb{R}^n$, satisfying the conditions:

1. Initial condition: $x(t_0) = x_0$.
2. Derivative: For all $t \in \mathbb{R} \setminus D$, $\frac{d}{dt}x(t) = A(t)x(t) + B(t)u(t)$.

Remark 2.17 Here, we take solutions to be defined for *all* $t \in \mathbb{R}$. We'll later see that this is justified for the case of LTV systems with piecewise continuous data. However, this requirement should be relaxed for *nonlinear* initial value problems.

Remark 2.18 Here, we assume that each map has the same discontinuity set - this assumption is made without loss of generality, since one can always take the union of the discontinuity sets of $A(\cdot), B(\cdot), u(\cdot)$ if they do not initially coincide.

This formal definition of a solution to an initial value problem *relaxes* our "intuitive" definition of a solution. It tells us that we only need to check the derivative condition at times when all data defining the initial value problem is continuous. Since carrying around the discontinuity set D can get a little cumbersome, we state an *equivalent* definition of a solution to an initial value problem which doesn't require the use of D. We state this definition in the form of a proposition.

Proposition 2.3 (Integral Solution of IVPs) *Consider the piecewise continuous maps $A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times n})$, $B(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times m})$, and $u \in PC(\mathbb{R}, \mathbb{R}^m)$. A function $x : \mathbb{R} \to \mathbb{R}^n$ is a solution to the initial value problem,*

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \ x(t_0) = x_0, \tag{2.49}$$

if and only if, for all $t \in \mathbb{R}$, it satisfies

$$x(t) = x_0 + \int_{t_0}^{t} A(\tau)x(\tau) + B(\tau)u(\tau)d\tau. \tag{2.50}$$

The proof of this proposition is essentially a straightforward application of the fundamental theorem of calculus, which we now recall.

Theorem 2.1 (Fundamental Theorem of Calculus) . *Let $f : \mathbb{R} \to \mathbb{R}^n$ be a Riemann-integrable function.*[2] *Then, the following results hold:*

1. *Fix a number $t_0 \in \mathbb{R}$ and define $F(t) = \int_{t_0}^{t} f(\tau)d\tau$. Then, F is continuous. Further, if f is continuous at t, then F is differentiable at t, with $F'(t) = f(t)$.*
2. *If $F : \mathbb{R} \to \mathbb{R}^n$ is a Riemann-integrable function satisfying $F'(t) = f(t)$ for all but a finite number of points in an interval $[t_0, t_1] \subseteq \mathbb{R}$, then $\int_{t_0}^{t_1} f(\tau)d\tau = F(t_1) - F(t_0)$.*

Proof See [1] for details. □

Now, we return to the proof of Proposition 2.3.

Proof (Of Proposition 2.3) Suppose $x : \mathbb{R} \to \mathbb{R}^n$ is a solution to the initial value problem in the sense of Definition 2.17. Then, for D the shared discontinuity set of $A(\cdot), B(\cdot), u(\cdot)$, one has that $\dot{x}(t) = A(t)x(t) + B(t)u(t)$, for all $t \in \mathbb{R} \setminus D$. We aim to show that x satisfies the integral condition proposed above.

Fix a time $t \in \mathbb{R}$, assuming $t \geq t_0$ (the proof is identical for $t < t_0$). By definition of a piecewise continuous function, it follows that $[t_0, t] \cap D$ only contains a finite number of

[2] We say that a function $f : \mathbb{R} \to \mathbb{R}^n$ is Riemann-integrable if its Riemann integral $\int_a^b f(t)dt$ is defined for all (finite) $a, b \in \mathbb{R}$. One may show that all piecewise continuous functions are Riemann-integrable.

points. Thus, between t_0 and t_1, $\dot{x}(t) = A(t)x(t) + B(t)u(t)$ at all but a finite number of points. By the fundamental theorem of calculus, it then follows that

$$x(t) = x(t_0) + \int_{t_0}^{t} A(\tau)x(\tau) + B(\tau)u(\tau)d\tau = x_0 + \int_{t_0}^{t} A(\tau)x(\tau) + B(\tau)u(\tau)d\tau. \tag{2.51}$$

This completes the first direction of the proof. Now, we proceed in the other direction. Suppose $x(t) = x_0 + \int_{t_0}^{t} A(\tau)x(\tau) + B(\tau)u(\tau)d\tau$ for all $t \geq t_0$. Taking $t = t_0$, one gets that $x(t_0) = x_0$, yielding the initial condition constraint. Now, we focus on differentiability. We know that $A(t)x(t) + B(t)u(t)$ must be continuous at all $t \in \mathbb{R} \setminus D$, since x is continuous by the fundamental theorem and $A(\cdot), B(\cdot), u(\cdot)$ are continuous on $\mathbb{R} \setminus D$. By the fundamental theorem, x is differentiable on $\mathbb{R} \setminus D$ and satisfies $\dot{x}(t) = A(t)x(t) + B(t)u(t)$ for all $t \in \mathbb{R} \setminus D$. We conclude that x is a solution to the initial value problem. □

2.2.1.2 A Matrix Initial Value Problem & The Peano-Baker Series

In the previous section, we wrote a formal definition for a solution to an initial value problem and formulated an equivalent definition of a solution using integration. In this subsection, we will examine a special, simple initial value problem that will provide almost *complete* insight into the existence and uniqueness problem we're aiming to solve.

Since the initial value problem $\dot{x}(t) = A(t)x(t) + B(t)u(t)$, $x(t_0) = x_0$ seems quite complex to analyze, having a number of moving parts, it might be beneficial to start with a simpler problem. Let's drop the input term, and study solutions to the initial value problem,

$$\dot{x}(t) = A(t)x(t),\ x(t_0) = x_0. \tag{2.52}$$

Can we simplify this problem even further? Using the integral definition of a solution to an initial value problem, we make the following observation.

Lemma 2.1 (Matrix IVP/Vector IVP) *Consider the initial value problem,*

$$\dot{x}(t) = A(t)x(t),\ x(t_0) = x_0. \tag{2.53}$$

If $X : \mathbb{R} \to \mathbb{R}^{n \times n}$ solves the matrix initial value problem,

$$\dot{X}(t) = A(t)X(t), X(t_0) = I, \tag{2.54}$$

then $x(t) = X(t)x_0$ solves the vector initial value problem $\dot{x}(t) = A(t)x(t)$, $x(t_0) = x_0$.

Remark 2.19 In this lemma, we use a solution to a *matrix* initial value problem. Solutions to such initial value problems are defined identically to vector initial value problems. In fact, we can write down a vector initial value problem corresponding to any given matrix initial value problem by stretching the matrix out into a vector. Because of this equivalence, the integral definition of a solution to an IVP still holds in the matrix case. Try writing down a formal definition of a solution to a matrix IVP to check your understanding!

Proof Suppose $X(t)$ solves the matrix IVP defined in the statement of the lemma. Then,

$$X(t) = I + \int_{t_0}^{t} A(\tau)X(\tau)d\tau. \tag{2.55}$$

Multiplying by x_0, one has,

$$X(t)x_0 = x_0 + \int_{t_0}^{t} A(\tau)X(\tau)x_0 d\tau, \tag{2.56}$$

which implies $x(t) = X(t)x_0$ is a solution to the IVP $\dot{x}(t) = A(t)x(t)$, $x(t_0) = x_0$. □

This lemma yields a great deal of insight into into the structure of solutions to $\dot{x}(t) = A(t)x(t),\ x(t_0) = x_0$. In particular, it tells us that there exist solutions to the initial value problem that are *linear* in the initial condition! Further, these solutions are *entirely* determined by solutions to the *matrix* IVP $\dot{X}(t) = A(t)X(t)$, $X(t_0) = I$. Thus, in order to understand solutions to the vector initial value problem $\dot{x}(t) = A(t)x(t),\ x(t_0) = x_0$, we will study solutions to the associated *matrix* initial value problem, $\dot{X}(t) = A(t)X(t)$, $X(t_0) = I$.

What do we know about solutions to this matrix IVP? Do solutions exist? If so, what form do they take? A solution to the matrix IVP must satisfy the integral equation,

$$X(t) = I + \int_{t_0}^{t} A(\tau)X(\tau)d\tau. \tag{2.57}$$

We notice that $X(\cdot)$ appears both on the left and right hand sides of the expression. Let's try re-plugging in the integral form of the solution into the $X(\tau)$ on the right hand side. This gives,

$$X(t) = I + \int_{t_0}^{t} A(\tau)\left[I + \int_{t_0}^{\tau} A(\tau')X(\tau')d\tau'\right]d\tau \tag{2.58}$$

$$= I + \left[\int_{t_0}^{t} A(\tau)d\tau\right] + \int_{t_0}^{t} A(\tau)\left[\int_{t_0}^{\tau} A(\tau')X(\tau')d\tau'\right]d\tau. \tag{2.59}$$

Interestingly, what we get inside the larger integral is the *same* expression that we originally substituted into. Thus, if we substitute again—this time for $X(\tau')$—we would find the same pattern! Indefinitely performing this substitution leads to the following definition.

Definition 2.18 (Peano-Baker Series) Let $A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times n})$. The Peano-Baker series with respect to $A(\cdot)$ is the infinite matrix series,

$$\Phi(t, t_0) = \sum_{k=0}^{\infty} S_k(t, t_0), \tag{2.60}$$

whose summands are defined by the recurrence,

$$S_0 = I,\ S_{k+1}(t, t_0) = \int_{t_0}^{t} A(\tau)S_k(\tau, t_0)d\tau. \tag{2.61}$$

In order to confirm that this definition is well-posed, we must, we must confirm that the Peano-Baker series actually converges. A useful tool for certifying the uniform convergence[3] of a series of functions is the *Weierstrass M-Test.*

[3] Recall from Chapter 1 that a sequence of functions $f_n : I \subseteq \mathbb{R} \to V$ (for $(V, \|\cdot\|)$ a normed vector space) converges *uniformly* if it converges with respect to the sup norm, $\|f\|_\infty = \sup_{t\in I} \|f(t)\|$.

Theorem 2.2 (Weierstrass M-Test) *Let $(V, \|\cdot\|)$ be a finite dimensional, normed vector space. Let $\{f_n\}$ be a collection of mappings $f_n : I \to V$, where $I \subseteq \mathbb{R}$. Let $\{M_n\} \subseteq \mathbb{R}$ be a sequence for which $\sup_{t\in I} \|f_n(t)\| \leq M_n$. If $\sum_{n=1}^{\infty} M_n$ converges, then $\sum_{n=1}^{\infty} f_n(t)$ converges uniformly on I.*

Proof See [1] for the details. □

In order to apply the Weierstrass M-test to certify the convergence of the Peano-Baker series, we require the following lemma.

Lemma 2.2 (Suprema of Piecewise Continuous Functions) *Consider a piecewise continuous function $f \in PC(I, \mathbb{R})$, where $I \subseteq \mathbb{R}$. For any compact subset $K \subseteq \mathbb{R}$, $\sup_{t\in K\cap I} f(t) < \infty$.*

Proof See Problem 2.2. □

With these results in mind, we study the convergence of the Peano-Baker series.

Proposition 2.4 (Convergence of Peano-Baker Series) *Let $A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times n})$ and $[-t', t'] \subseteq \mathbb{R}$, $t' \geq 0$, be a finite, closed interval. For each fixed $t_0 \in [-t', t']$, the Peano-Baker series $\Phi(\cdot, t_0)$ defined by $A(\cdot)$ converges uniformly on $[-t', t']$.*

Proof Our proof follows the Weierstrass M-test. Fix an interval $[-t', t'] \subseteq \mathbb{R}$ and an initial time $t_0 \in [-t', t']$. Recall that the Peano-Baker series at time $t \in [-t', t']$ is defined,

$$\Phi(t, t_0) = \sum_{k=0}^{\infty} S_k(t, t_0),\ S_0 = I,\ S_k(t, t_0) = \int_{t_0}^{t} A(\tau)S_{k-1}(\tau, t_0)d\tau, \tag{2.62}$$

In order to prove that this series converges uniformly using the Weierstrass M-test, we'll exhibit a uniform bound on each $S_k(\cdot, t_0)$ over the first argument. First, we'll show that the bound,

$$\|S_k(t, t_0)\| \leq \frac{1}{k!}\Big(\sup_{t\in[-t',t']} \|A(t)\|\Big)^k |t - t_0|^k,\ \forall t \in [-t', t'], \tag{2.63}$$

must hold. Notice that $\sup_{t\in[-t',t']} \|A(t)\|$ is finite by Lemma 2.2. Note that—although it seems like we're pulling this bound out of thin air—this is actually something one can discover by playing around with the first few terms of the series. You're encouraged to try this if you're not convinced!

Let's prove that the bound holds by induction on k. The base case, $k = 0$, is trivial. We get $\|S_0\| = \|I\| = 1$, which matches the proposed bound exactly. Let $k \geq 1$, and assume for induction that the proposed bound holds. Now, we bound S_{k+1} for arbitrary t, t_0—keep in mind, we may have $t \geq t_0$ or $t < t_0$. We have,

$$\begin{aligned}\|S_{k+1}(t, t_0)\| &= \left\|\int_{t_0}^{t} A(\tau)S_k(\tau, t_0)d\tau\right\| &(2.64)\\ &\leq \left|\int_{t_0}^{t} \|A(\tau)\|\,\|S_k(\tau, t_0)\|\,d\tau\right| &(2.65)\\ &\leq \left|\int_{t_0}^{t} \sup_{\tau\in[-t',t']} \|A(\tau)\| \cdot \frac{1}{k!}\Big(\sup_{\tau\in[-t',t']} \|A(\tau)\|\Big)^k |\tau - t_0|^k d\tau\right|. &(2.66)\end{aligned}$$

Now, we split into two cases. First, assume $t \geq t_0$. In this case, we have that the above is bounded,

$$\|S_{k+1}(t, t_0)\| \leq \frac{1}{k!}\Big(\sup_{t\in[-t',t']} \|A(t)\|\Big)^{k+1} \left| \int_{t_0}^{t} |\tau - t_0|^k d\tau \right| \tag{2.67}$$

$$= \frac{1}{k!}\Big(\sup_{t\in[-t',t']} \|A(t)\|\Big)^{k+1} \frac{1}{k+1} |t - t_0|^{k+1} \tag{2.68}$$

$$= \frac{1}{(k+1)!}\Big(\sup_{t\in[-t',t']} \|A(t)\|\Big)^{k+1} |t - t_0|^{k+1}, \ \forall t \in [t_0, t_1]. \tag{2.69}$$

Thus, the proposed bound holds for $t \geq t_0$. For $t < t_0$, the same procedure is followed—simply flip t_0 and t in the integral and perform the same bounds. So, by induction on k, we conclude that the proposed bound holds for all $k \in \mathbb{N}$. We can then bound $\|S_k(t, t_0)\|$ uniformly in t on $[-t', t']$ by,

$$\sup_{t\in[-t',t']} \|S_k(t, t_0)\| \leq \frac{1}{k!}\Big(\sup_{t\in[-t,t]} \|A(t)\|\Big)^{k} (2t')^k. \tag{2.70}$$

Now, we're ready to apply the Weierstrass M-test. Define M_k as the right hand side of the inequality above. Does $\sum_{k=0}^{\infty} M_k$ converge? We recognize the sum as the Taylor series definition of the *exponential*! Thus, we have,

$$\sum_{k=0}^{\infty} \frac{1}{k!}\Big(\sup_{t\in[-t',t']} \|A(t)\|\Big)^{k} (2t')^k = \exp\Big(\sup_{t\in[-t',t']} \|A(t)\| \, (2t')\Big) < \infty. \tag{2.71}$$

By the Weierstrass M-test, we conclude that the Peano-Baker series converges uniformly on any compact interval $[-t', t']$. □

Let's summarize what we've done so far. In analyzing solutions to the matrix IVP, $\dot{X}(t) = A(t)X(t)$, $X(t_0) = I$, we discovered a recurrent pattern that led us to the Peano-Baker series. Then, we proved that the Peano-Baker series converges uniformly on any compact interval. Now, we ask the question—does it converge to the solution of the matrix IVP? The following theorem provides an answer.

Proposition 2.5 (Peano-Baker Series Solves the Matrix IVP) *Let $A \in PC(\mathbb{R}, \mathbb{R}^{n\times n})$. Consider the matrix initial value problem $\dot{X}(t) = A(t)X(t)$, $X(t_0) = I$. The Peano-Baker series $\Phi(t, t_0)$ defined by $A(\cdot)$ solves the matrix initial value problem.*

Proof We can prove this either via direct differentiation or the integral method. Let's proceed via differentiation. Let $D \subseteq \mathbb{R}$ be the discontinuity set of A. By definition, one has that for all $t \in \mathbb{R} \setminus D$, each term of the Peano-Baker series is differentiable in its first argument. Thus, for any time $t \in \mathbb{R}$,

$$\begin{aligned}
\frac{d}{dt}\sum_{k=0}^{n} S_k(t,t_0) &= \sum_{k=0}^{n} \frac{d}{dt} S_k(t,t_0) && (2.72)\\
&= \sum_{k=1}^{n} \frac{d}{dt}\int_{t_0}^{t} A(\tau)S_{k-1}(\tau,t_0)d\tau && (2.73)\\
&= A(t)\sum_{k=1}^{n} S_{k-1}(t,t_0) && (2.74)\\
&= A(t)\sum_{k=0}^{n-1} S_k(t,t_0). && (2.75)
\end{aligned}$$

Now, fix an interval $[-t_1, t_1] \subseteq \mathbb{R}$, for which $t_0 \in [-t_1, t_1]$. Since the Peano-Baker series converges uniformly on this interval, it follows that $\lim_{n\to\infty} \frac{d}{dt}\sum_{k=0}^{n} S_k(t,t_0) = \lim_{n\to\infty} \sum_{k=0}^{n} \frac{d}{dt} S_k(t,t_0)$, for all $t \in (-t_1, t_1) \cap (\mathbb{R} \setminus D)$. This means,

$$\frac{d}{dt}\sum_{k=0}^{\infty} S_k(t,t_0) = A(t)\lim_{n\to\infty}\sum_{k=0}^{n-1} S_k(t,t_0) = A(t)\sum_{k=0}^{\infty} S_k(t,t_0), \tag{2.76}$$

for all $t \in (-t_1, t_1) \cap (\mathbb{R} \setminus D)$. Thus, we have that,

$$\frac{d}{dt}\Phi(t,t_0) = A(t)\Phi(t,t_0),\ \forall t \in (-t_1, t_1) \cap (\mathbb{R} \setminus D). \tag{2.77}$$

Since the interval $[-t_1, t_1]$ can be made arbitrarily large, we conclude that the Peano-Baker series satisfies the derivative property of the initial value problem for all $t \in \mathbb{R} \setminus D$. Further, we have that $\Phi(t_0, t_0) = I$ by definition. We conclude that the Peano-Baker series solves the matrix initial value problem. □

We've now established that the Peano-Baker series is *a* solution to the matrix initial value problem—is it the *only* solution? The following inequality helps us answer this question.

Lemma 2.3 (Gronwall Inequality) *Let $y, k \in PC(\mathbb{R}, \mathbb{R}_{\geq 0})$ and $c \in \mathbb{R}_{\geq 0}$, and $t_0 \in \mathbb{R}$. If for all $t \in \mathbb{R}$, y satisfies,*

$$y(t) \leq c + \left|\int_{t_0}^{t} k(\tau)y(\tau)d\tau\right|, \tag{2.78}$$

then for all $t \in \mathbb{R}$,

$$y(t) \leq c\exp\left|\int_{t_0}^{t} k(\tau)d\tau\right|. \tag{2.79}$$

Proof See Problem 2.5. □

Using the Gronwall inequality, we prove that solutions to the matrix IVP are *unique.*

Theorem 2.3 (Existence & Uniqueness of Solutions to Matrix IVP) *Let $A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times n})$. The Peano-Baker series, $\Phi(t, t_0)$, is the unique solution to the matrix initial value problem $\dot{X}(t) = A(t)X(t),\ X(t_0) = I$.*

Proof By Proposition 2.5, we already know $\varPhi(t,t_0)$ is a solution the IVP. Now, we show it is the unique solution. Suppose $X:\mathbb{R}\to\mathbb{R}^{n\times n}$ is another solution. Then, for any t,t_0, both X and $\varPhi$ must satisfy,

$$X(t) = I + \int_{t_0}^{t} A(\tau)X(\tau)d\tau \tag{2.80}$$

$$\varPhi(t,t_0) = I + \int_{t_0}^{t} A(\tau)\varPhi(\tau,t_0)d\tau. \tag{2.81}$$

Subtracting and taking the norm, we get,

$$\begin{aligned}\|\varPhi(t,t_0)-X(t)\| &= \left\|\int_{t_0}^{t} A(\tau)(\varPhi(\tau,t_0)-X(\tau))d\tau\right\| &(2.82)\\ &\leq \left|\int_{t_0}^{t} \|A(\tau)\|\,\|\varPhi(\tau,t_0)-X(\tau)\|\,d\tau\right|. &(2.83)\end{aligned}$$

Applying the Gronwall lemma, we find that

$$\|\varPhi(t,t_0)-X(t)\| = 0,\ \forall t\in\mathbb{R}. \tag{2.84}$$

We conclude that $\varPhi(t,t_0)=X(t)$ for all $t\in\mathbb{R}$, and that solutions to the IVP are unique. □

2.2.1.3 The State Transition Matrix

In the previous subsection, we developed the theory of the *Peano-Baker series* to prove the existence of a unique solution to the matrix initial value problem,

$$\dot{X}(t) = A(t)X(t),\ X(t_0) = I. \tag{2.85}$$

Since the integral formula for the Peano-Baker series is rather impractical to work with, we'll find it convenient to *abstract away* the computation of the Peano-Baker series and focus on $\varPhi(t,t_0)$ as the solution to the matrix initial value problem. In this spirit, we make the following definition.

Definition 2.19 (State Transition Matrix) Let $A(\cdot)\in PC(\mathbb{R},\mathbb{R}^{n\times n})$. The state transition matrix with respect to $A(\cdot)$ is a map $\varPhi:\mathbb{R}\times\mathbb{R}\to\mathbb{R}^{n\times n}$, such that $\varPhi(\cdot,t_0):\mathbb{R}\to\mathbb{R}^{n\times n}$ is the unique solution to the initial value problem,

$$\dot{X}(t) = A(t)X(t),\ X(t_0) = I. \tag{2.86}$$

Remark 2.20 Despite its name, the state transition *matrix* is not a fixed matrix but rather a *map* into the set of matrices.

We emphasize—the state transition matrix $\varPhi(t,t_0)$ is *exactly* calculated by the Peano-Baker series. Here, we simply hide the Peano-Baker series behind a layer of abstraction—the *state transition matrix*—to emphasize that we don't want to use the series as an analysis tool.

By focusing on the "abstracted" definition of $\varPhi(t,t_0)$ as the unique solution of a differential equation, as opposed to the definition of $\varPhi(t,t_0)$ as an infinite series, we'll find that we can write much more elegant proofs.

Proposition 2.6 (Properties of the State Transition Matrix) *Let $A(\cdot) \in \mathbb{P}C(\mathbb{R}, \mathbb{R}^{n\times n})$. The state transition matrix Φ with respect to $A(\cdot)$ satisfies the following properties:*

1. *Composition: For all $t_0, t_1, t_2 \in \mathbb{R}$, $\Phi(t_2, t_0) = \Phi(t_2, t_1)\Phi(t_1, t_0)$.*
2. *Inverse: For all $t_0, t_1 \in \mathbb{R}$, $\Phi(t_1, t_0)$ is invertible with $[\Phi(t_1, t_0)]^{-1} = \Phi(t_0, t_1)$.*

Proof First, we show the composition property. To prove this, we will use the uniqueness property of solutions to the initial value problem, $\dot{X}(t) = A(t)X(t),\ X(t_0) = X_0$, which follows from the Gronwall Lemma. In particular, we will show that, for all $t_0, t_1 \in \mathbb{R}$, both

$$\Phi(\cdot, t_0) : \mathbb{R} \to \mathbb{R}^{n\times n} \tag{2.87}$$

$$\Phi(\cdot, t_1)\Phi(t_1, t_0) : \mathbb{R} \to \mathbb{R}^{n\times n}, \tag{2.88}$$

are solutions to the matrix IVP $\dot{X}(t) = A(t)X(t),\ X(t_1) = \Phi(t_1, t_0)$. Then, we'll use uniqueness to conclude that they are equal. First, we know that $\Phi(\cdot, t_0)$ is a solution to the matrix IVP by definition of the state transition matrix. Thus, $\Phi(\cdot, t_0)$ satisfies,

$$\frac{d}{dt}\Phi(t, t_0) = A(t)\Phi(t, t_0)\ \forall t \in \mathbb{R} \setminus D. \tag{2.89}$$

where D is the discontinuity set of A. This implies that $\Phi(t, t_0)$ is also the solution the matrix IVP, $\dot{X}(t) = A(t)X(t),\ X(t_1) = \Phi(t_1, t_0)$. Now, we check the same for the second. We have that, for D the discontinuity set of $A(\cdot)$,

$$\frac{d}{dt}[\Phi(t, t_1)\Phi(t_1, t_0)] = A(t)\Phi(t, t_1)\Phi(t_1, t_0) = A(t)[\Phi(t, t_1)\Phi(t_1, t_0)],\ \forall t \in \mathbb{R} \setminus D. \tag{2.90}$$

Further, we have that $\Phi(t_1, t_1)\Phi(t_1, t_0) = I\Phi(t_1, t_0) = \Phi(t_1, t_0)$. Therefore, $\Phi(\cdot, t_1)\Phi(t_1, t_0)$ *also* solves the initial value problem! By the Gronwall Lemma, it follows that solutions to the IVP are unique, which implies,

$$\Phi(t_2, t_1)\Phi(t_1, t_0) = \Phi(t_2, t_0),\ \forall t_0, t_1, t_2 \in \mathbb{R}. \tag{2.91}$$

This completes the proof of the first item. The second item follows by direct application of the first. Fix times $t_0, t_1 \in \mathbb{R}$. Then, it follows from the composition rule that

$$\Phi(t_0, t_1)\Phi(t_1, t_0) = \Phi(t_0, t_0) = I \tag{2.92}$$

$$\Phi(t_1, t_0)\Phi(t_0, t_1) = \Phi(t_1, t_1) = I. \tag{2.93}$$

So, we conclude by the uniqueness of the matrix inverse that $\Phi(t_0, t_1) = [\Phi(t_1, t_0)]^{-1}$. □

2.2.1.4 The Continuous-Time, LTV Initial Value Problem

We're finally ready to tackle our original problem: proving the existence & uniqueness of solutions to the initial value problem,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(t_0) = x_0. \tag{2.94}$$

Amazingly, *all we need* to construct solutions to this problem is the state transition matrix. To determine a formula for the solutions to such initial value problems, we'll need the differentiation under the integral rule, which we now recall.

Theorem 2.4 (Leibniz Rule for Differentiation Under the Integral) *Let $f \in C^1(\mathbb{R}\times\mathbb{R},\mathbb{R}^n)$ and $a(\cdot), b(\cdot) \in C^1(\mathbb{R},\mathbb{R})$ be continuously differentiable functions. For all $t \in \mathbb{R}$,*

$$\frac{d}{dt}\Big[\int_{a(t)}^{b(t)} f(t,\tau)d\tau\Big] = \int_{a(t)}^{b(t)} \frac{\partial}{\partial t} f(t,\tau)d\tau + f(t,b(t))\frac{d}{dt}b(t) - f(t,a(t))\frac{d}{dt}a(t). \tag{2.95}$$

Proof See Problem 2.6 for details. □

With this in mind, we state a theorem on the existence and uniqueness of solutions to the LTV initial value problem.

Theorem 2.5 (Existence & Uniqueness of Solutions to LTV-IVP) *Let $A(\cdot) \in PC(\mathbb{R},\mathbb{R}^{n\times n})$, $B(\cdot) \in PC(\mathbb{R},\mathbb{R}^{n\times m})$, and $u(\cdot) \in PC(\mathbb{R},\mathbb{R}^m)$. The unique solution to the initial value problem,*

$$\dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(t_0) = x_0, \tag{2.96}$$

is given by the map $x : \mathbb{R} \to \mathbb{R}^n$, defined,

$$x(t) = \Phi(t,t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)d\tau, \tag{2.97}$$

where Φ is the state transition matrix with respect to $A(\cdot)$.

Proof Our proof follows by direct differentiation. Let $t \in \mathbb{R} \setminus D$, for D the shared discontinuity set of $A(\cdot)$, $B(\cdot)$, and $u(\cdot)$. By definition of the state transition matrix, it follows that

$$\begin{aligned}
\frac{d}{dt}x(t) &= \frac{d}{dt}\Phi(t,t_0)x_0 + \frac{d}{dt}\int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)d\tau && (2.98)\\
&= A(t)\Phi(t,t_0)x_0 + \Phi(t,t)B(t)u(t) + \int_{t_0}^{t} \frac{\partial}{\partial t}\Phi(t,\tau)B(\tau)u(\tau)d\tau && (2.99)\\
&= A(t)\Phi(t,t_0)x_0 + B(t)u(t) + \int_{t_0}^{t} A(t)\Phi(t,\tau)B(\tau)u(\tau)d\tau && (2.100)\\
&= A(t)\Big[\Phi(t,t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)d\tau\Big] + B(t)u(t) && (2.101)\\
&= A(t)x(t) + B(t)u(t). && (2.102)
\end{aligned}$$

Thus, the solution satisfies the differentiation property. Also by definition of the state transition matrix,

$$x(t_0) = \Phi(t_0,t_0)x_0 + \int_{t_0}^{t_0} \Phi(t_0,\tau)B(\tau)u(\tau)d\tau = Ix_0 + 0 = x_0. \tag{2.103}$$

Therefore, the initial condition is also satisfied. This establishes the *existence* of solutions to the initial value problem. Now, we verify uniqueness using the Gronwall inequality. Suppose

$\hat{x} : \mathbb{R} \to \mathbb{R}^n$ is another solution to the initial value problem. Then, both x and $\hat{x}$ must satisfy,

$$x(t) = x_0 + \int_{t_0}^{t} A(\tau)x(\tau) + B(\tau)u(\tau)d\tau \tag{2.104}$$

$$\hat{x}(t) = x_0 + \int_{t_0}^{t} A(\tau)\hat{x}(\tau) + B(\tau)u(\tau)d\tau,\ \forall t \in \mathbb{R}. \tag{2.105}$$

Subtracting and taking the norm, one gets,

$$\|x(t) - \hat{x}(t)\| = \left\| \int_{t_0}^{t} A(\tau)(x(\tau) - \hat{x}(\tau)d\tau \right\| \tag{2.106}$$

$$\leq \left| \int_{t_0}^{t} \|A(\tau)\| \, \|x(\tau) - \hat{x}(\tau)\| \, d\tau \right|. \tag{2.107}$$

Applying the Gronwall inequality, it follows that $\|x(t) - \hat{x}(t)\| = 0$, for all $t \in \mathbb{R}$. We conclude that $x = \hat{x}$, and that solutions to the IVP are unique. □

This result yields the final piece in the puzzle in establishing that a continuous-time, linear time-varying system representation yields a continuous-time, linear I/O dynamical system. Since the existence and uniqueness theorem above does the bulk of the work, we leave the details of the following theorem to the reader.

Theorem 2.6 (LTV System Representations Determine Linear I/O Systems) *Consider a continuous-time, LTV system representation $(A(\cdot), B(\cdot), C(\cdot), D(\cdot))$. This representation determines a linear I/O system $\mathcal{D}$, comprised of the following data:*

1. *Time Set: $\mathcal{T} = \mathbb{R}$.*
2. *Spaces: $\mathcal{U} = PC(\mathbb{R}, \mathbb{R}^m)$, $\mathcal{Y} = PC(\mathbb{R}, \mathbb{R}^p)$, and $\Sigma = \mathbb{R}^n$.*
3. *State Transition Map: the state transition map is computed,*

$$\varphi(t_1, t_0, x_0, u(\cdot)) = \Phi(t_1, t_0)x_0 + \int_{t_0}^{t_1} \Phi(t_1, \tau)B(\tau)u(\tau)d\tau. \tag{2.108}$$

4. *Readout Map: the readout map is computed,*

$$r(t, x, u) = C(t)x + D(t)u. \tag{2.109}$$

5. *I/O Map: the I/O map is computed,*

$$\rho(t_1, t_0, x_0, u(\cdot)) = C(t_1)\Phi(t_1, t_0)x_0 + C(t_1)\int_{t_0}^{t_1} \Phi(t_1, \tau)B(\tau)u(\tau)d\tau + D(t_1)u(t_1). \tag{2.110}$$

Proof See Problem 2.7. □

To conclude, we make the following important observation. The evolution of the output of any continuous-time, LTV system representation can be *decomposed* into the sum of the zero-input and zero-state components:

$$\rho(t_1, t_0, x_0, u(\cdot)) = \underbrace{C(t_1)\Phi(t_1, t_0)x_0}_{\text{Zero-Input Response}} + \underbrace{C(t_1)\int_{t_0}^{t_1} \Phi(t_1, \tau)B(\tau)u(\tau)d\tau + D(t_1)u(t_1)}_{\text{Zero-State Response}}. \quad (2.111)$$

Note that the zero-input and zero-state responses are sometimes referred to as the *free* and *forced* responses, respectively. This tells us that the response of any linear, time-varying system to an input signal has a component due to the initial condition and a separate component due to the input.

2.2.2 Solutions of Discrete-Time Linear Systems

After taking on a monumental challenge in the desert of continuous-time systems, it's now time to relax in the oasis of discrete-time systems. Kick back, grab your favorite normed vector space, and prepare to be relieved by a *substantially* easier theory.

In this section, we'll work through the process of proving that discrete-time, linear time-varying systems define discrete-time linear I/O systems. Why is the theory in the discrete-time case so much easier than in the continuous-time case? Let's take a quick look at the state equation for the discrete-time, linear time-varying system and see what we find. We have,

$$x[k+1] = A[k]x[k] + B[k]u[k]. \quad (2.112)$$

That is, given $x[k]$ and $u[k]$, we can immediately calculate $x[k+1] = A[k]x[k] + B[k]u[k]$. This means that, for any initial condition $x[k_0] = x_0$ and input signal $u[\cdot] : \mathbb{Z} \to \mathbb{R}^m$, we can recursively solve for $x[k],\ k \geq k_0$. This makes the problem of existence and uniqueness of solutions trivial in discrete-time.

Despite this great simplification of the discrete-time initial value problem, we'll still find it fruitful to examine closely the structure of solutions to a discrete-time system—just because we can write down a solution directly from the recurrence doesn't mean there isn't more at play! Interestingly, we'll find that a state transition matrix similar to that of the continuous-time case also appears in the discrete-time case.

In the remainder of this section, we'll follow the same general procedure as in the continuous-time case. Here, we'll leave many of the results as exercises or problems, due to their simpler analytical nature.

2.2.2.1 Defining Solutions to Discrete-Time Systems

As with the continuous-time case, we begin by specifying a formal definition of a solution to a discrete-time initial value problem. In this case, since the time set is discrete and the state equation is a recurrence relation, we won't need to worry about regularity conditions such as piecewise continuity. This is reflected in the simpler form of the definition.

Definition 2.20 (Solution to Discrete-Time Recurrence) Let $A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n}$, $B[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times m}$, and $u[\cdot] : \mathbb{Z} \to \mathbb{R}^m$ be sequences. For $x_0 \in \mathbb{R}^n$ and $k_0 \in \mathbb{Z}$, a solution to the discrete-time recurrence,

$$x[k+1] = A[k]x[k] + B[k]u[k],\ x[k_0] = x_0, \tag{2.113}$$

is a sequence $x[\cdot] : \mathbb{Z}_{\geq k_0} \to \mathbb{R}^n$, satisfying:

1. Initial condition: $x[k_0] = x_0$.
2. Recurrence: For all $k \geq k_0$, $x[k+1] = A[k]x[k] + B[k]u[k]$.

We observe that the solution to a discrete-time initial value problem has exactly the structure we expect! It's important to note that—instead of the solution being defined on all of $\mathbb{Z}$, solutions are defined as sequences starting at a time k_0.

The reasoning behind this is essentially as follows: for $k < k_0$, it's possible that the trajectory leading to $x[k_0] = x_0$ is *not* uniquely defined (can you think of the reason why? We'll provide an answer below). For these reasons, we restrict the definition of a solution to be for $k \geq k_0$. Shortly, we'll describe some conditions that let us extend the domain of the definition of a solution to all of $\mathbb{Z}$.

2.2.2.2 A Matrix Initial Value Problem & The State Transition Matrix

In order to uncover the structure underlying the discrete-time initial value problem, we again take the approach of studying a *matrix* initial value problem. Fortunately, as mentioned above, the existence and uniqueness of solutions are no longer a concern! As such, we can directly jump to the definition of a state transition matrix.

Definition 2.21 (Discrete-Time State Transition Matrix) Consider a sequence $A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n}$. The (discrete-time) state transition matrix with respect to $A[\cdot]$ is a map

$$\Phi[\cdot,\cdot] : \mathbf{T} \to \mathbb{R}^{n\times n},\ \mathbf{T} := \{(k, k_0) \in \mathbb{Z}\times\mathbb{Z} : k \geq k_0\}, \tag{2.114}$$

such that for all $k_0 \in \mathbb{Z}$, $\Phi[\cdot, k_0]$ is the solution to the discrete-time, matrix initial value problem

$$X[k+1] = A[k]X[k],\ X[k_0] = I. \tag{2.115}$$

Thus, we define the state transition matrix Φ in *exactly* the same way as for the continuous-time case—as a solution to a matrix initial value problem defined by $A[\cdot]$, with an initial condition given by the identity matrix. Here, however, in order to get a well-defined solution, we must define the domain of Φ such that $k \geq k_0$. We'll see shortly how this assumption on the domain can be relaxed when the sequence $\{A[k]\}_{k\in\mathbb{Z}}$ has all nonsingular elements.

Proposition 2.7 (Structure of the Discrete-Time State Transition Matrix) *Consider a sequence $A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n}$. For any $k_0 \in \mathbb{Z}$ and $k \geq k_0$, the state transition matrix $\Phi[k, k_0]$ with respect to $A[\cdot]$ is computed*

$$\Phi[k_0, k_0] = I \tag{2.116}$$

$$\Phi[k+1, k_0] = A[k]\Phi[k, k_0]. \tag{2.117}$$

Exercise 2.6 Prove Proposition 2.7.

This result gives a method of calculating the state transition matrix for a given k_0 and all $k \geq k_0$. Why can't we calculate the state transition matrix for all $k \in \mathbb{Z}$? In the event where the matrix $A[k]$ is not invertible, we lose uniqueness in the definition of Φ—thus, for $k < k_0$, Φ might be ill-defined. In the event where $A[k]$ is nonsingular (i.e. invertible), however, we can make the following conclusion.

Proposition 2.8 (State Transition Matrix for Invertible A) *Consider a sequence $A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n}$, in which each $A[k]$ is nonsingular. For such a sequence, the state transition matrix Φ can be uniquely defined on all of $\mathbb{Z} \times \mathbb{Z}$.*

Exercise 2.7 Prove Proposition 2.8. What can go wrong if $A[k]$ is singular? Provide an example.

Finally, we show that the discrete-time state transition matrix satisfies a composability property. Here, due to the risk of singular $A[k]$, we *cannot* prove a composability property for all k_0, k_1, k_2—we are restricted to $k_0 \leq k_1 \leq k_2$. For this same reason, we are not guaranteed that the discrete-time state transition matrix is invertible for all $k_0, k_1 \in \mathbb{Z}$.

Proposition 2.9 (Composability of the Discrete-Time State Transition Matrix) *Consider a sequence $A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n}$. The state transition matrix Φ with respect to $A[\cdot]$ then satisfies $\Phi[k_2, k_0] = \Phi[k_2, k_1]\Phi[k_1, k_0]$, for all $k_0 \leq k_1 \leq k_2 \in \mathbb{Z}$.*

2.2.2.3 Solutions to the Discrete-Time, LTV Recurrence

Using the state transition matrix, we can find the unique solution to the discrete-time recurrence defined by the state equation of the discrete-time, LTV representation.

Theorem 2.7 (Solutions to Discrete-Time, LTV Recurrence) *Let $A[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times n}$, $B[\cdot] : \mathbb{Z} \to \mathbb{R}^{n\times m}$, and $u[\cdot] : \mathbb{Z} \to \mathbb{R}^m$ be sequences. For $x_0 \in \mathbb{R}^n$ and $k_0 \in \mathbb{Z}$, the unique solution to the discrete-time recurrence,*

$$x[k+1] = A[k]x[k] + B[k]u[k],\ x[k_0] = x_0, \tag{2.118}$$

is given by the sequence sequence $x[\cdot] : \mathbb{Z}_{\geq k_0} \to \mathbb{R}^n$, defined,

$$x[k] = \Phi[k, k_0]x_0 + \sum_{j=k_0}^{k-1} \Phi[k, j+1]B[j]u[j]. \tag{2.119}$$

Exercise 2.8 Prove Theorem 2.7 by induction on k. Why is the expression $\Phi[k, j+1]$ well-defined for $j \in [k_0, k-1] \cap \mathbb{Z}$?

Finally, we confirm that discrete-time, linear time-varying representations determine discrete-time linear I/O systems.

Theorem 2.8 (DT-LTV System Representations Determine DT Linear I/O Systems) *Consider a discrete-time, LTV system representation $(A[\cdot], B[\cdot], C[\cdot], D[\cdot])$. This representation determines a discrete-time linear I/O system $\mathcal{D}$, comprised of the data:*

1. *Time Set: $\mathcal{T} = \mathbb{Z}$.*
2. *Spaces: $\mathcal{U} = \{u : \mathbb{Z} \to \mathbb{R}^m\}$, $\mathcal{Y} = \{y : \mathbb{Z} \to \mathbb{R}^p\}$, and $\Sigma = \mathbb{R}^n$.*

3. State Transition Map: the state-transition map is computed,

$$\varphi(k_1, k_0, x_0, u[\cdot]) = \Phi[k_1, k_0]x_0 + \sum_{j=k_0}^{k_1-1} \Phi[k_1, j+1]B[j]u[j] \tag{2.120}$$

4. Readout Map: the readout map is computed,

$$r(k, x, u) = C[k]x + D[k]u \tag{2.121}$$

5. I/O Map: The I/O map is computed,

$$\rho(k_1, k_0, x_0, u[\cdot]) = C[k_1]\Phi[k_1, k_0]x_0 + C[k_1]\sum_{j=k_0}^{k_1-1} \Phi[k_1, j+1]B[j]u[j] + D[k_1]u[k_1]. \tag{2.122}$$

Proof See Problem 2.7. □

As with the case of a continuous-time linear system, we note that the state response of any discrete-time linear, time-varying system representation can be decomposed as the sum,

$$\rho(k_1, k_0, x_0, u[\cdot]) = \underbrace{C[k_1]\Phi[k_1, k_0]x_0}_{\text{Zero-Input Response}} + \underbrace{C[k_1]\sum_{j=k_0}^{k_1-1} \Phi[k_1, j+1]B[j]u[j] + D[k_1]u[k_1]}_{\text{Zero-State Response}}, \tag{2.123}$$

of a zero-input and a zero-state response. As with the continuous-time case, these components are also referred to as the *free* and *forced* responses, respectively.

2.2.3 Further Reading

This section was mainly influenced by [6], [20], and [22]. For an approach to the existence & uniqueness problem that uses a more general existence & uniqueness theorem for differential equations, the interested reader is encouraged to consult [16]. For a treatment of existence & uniqueness of solutions to differential equations with *measurable* data (more general than piecewise continuous), a measure-theoretic treatment of ordinary differential equations is found in Appendix C of [29].

2.2.4 Problems

Problem 2.3 (Transition Matrix Under Change of Variable) Consider a continuous-time linear, time-varying system representation $(A(\cdot), B(\cdot), C(\cdot), D(\cdot))$,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{2.124}$$

$$y(t) = C(t)x(t) + D(t)u(t). \tag{2.125}$$

1. Consider an invertible linear transformation $T \in \mathbb{R}^{n\times n}$ and a corresponding change of variables, $z = Tx$. Identify the system representation $(\hat{A}(\cdot), \hat{B}(\cdot), \hat{C}(\cdot), \hat{D}(\cdot))$ for which solutions to,

$$\dot{z}(t) = \hat{A}(t)\hat{z}(t) + \hat{B}(t)u(t) \tag{2.126}$$

$$\hat{y}(t) = \hat{C}(t)\hat{z}(t) + \hat{D}(t)u(t) \tag{2.127}$$

   satisfy $z(t) = Tx(t)$ and $\hat{y}(t) = y(t)$ for all initial conditions x_0 and Tx_0 and piecewise continuous input signals $u(\cdot)$. Conclude that the input to output behavior of the system *does not* depend on changes of state coordinates.
2. Write the state transition matrix $\hat{\Phi}(t, t_0)$ of the transformed system in terms of the state transition matrix $\Phi(t, t_0)$ of the original system and the transformation T.
3. Does the relation you derived in part (2) also hold for a discrete-time system representation? Explain why or why not.

Problem 2.4 (The Intermediate Value Property of the Derivative) In this problem, we formalize the "intermediate value property" of the derivative, which states that the derivative of a function cannot have any jump discontinuities. Recall that the derivative of a scalar function $f : \mathbb{R} \to \mathbb{R}$ is defined,

$$f'(t) = \lim_{\tau\to t} \frac{f(t) - f(\tau)}{t - \tau}. \tag{2.128}$$

It is *not* necessarily the case that the derivative of a differentiable function is continuous! However, we *can* exclude certain types of discontinuities.

1. First, suppose $f : \mathbb{R} \to \mathbb{R}$ is differentiable on an open interval (a, b). Show that if f attains a maximum or minimum value at a point $c \in (a, b)$, then $f'(c) = 0$.
2. Now, suppose $f : \mathbb{R} \to \mathbb{R}$ is differentiable on an open set $A \subseteq \mathbb{R}$ containing an interval $[a, b]$. Suppose $\alpha \in \mathbb{R}$ satisfies $f'(a) < \alpha < f'(b)$. Show there exists a point $c \in (a, b)$ for which $f'(c) = \alpha$. *If you get stuck, consult Chapter 5.2 of [1] for some hints.*
3. Apply the conclusion of part (2) to study the jump discontinuities of the derivative of a function $f : \mathbb{R} \to \mathbb{R}^n$.

Problem 2.5 (Gronwall Inequality) In this problem, we'll walk through a proof of the Gronwall inequality (Lemma 2.3). Recall that the Gronwall inequality is formulated as follows. Let $y, k \in PC(\mathbb{R}, \mathbb{R}_{\geq 0})$ and $\mu \in PC(R, \mathbb{R}_{\geq 0})$, $c \in \mathbb{R}_{\geq 0}$, and $t_0 \in \mathbb{R}$. If for all $t \in \mathbb{R}$, y satisfies,

$$y(t) \leq c + \left| \int_{t_0}^{t} k(\tau)y(\tau)d\tau \right|, \tag{2.129}$$

then for all $t \in \mathbb{R}$,

$$y(t) \leq c \exp \left| \int_{t_0}^{t} k(\tau)d\tau \right|. \tag{2.130}$$

Let's get to work on assembling a proof of this result.

1. Fix times $t, t_0 \in \mathbb{R}$ with $t > t_0$. Define a function,

$$Y(t) = c + \int_{t_0}^{t} k(\tau)y(\tau)d\tau. \tag{2.131}$$

Argue that $y(t) \leq Y(t)$ for all $t \geq t_0$, and that $Y(t)$ satisfies $\frac{d}{dt}Y(t) = k(t)y(t)$.

2. Prove that,

$$y(t) \leq Y(t)k(t)\exp(-\int_{t_0}^{t} k(\tau)d\tau), \tag{2.132}$$

and that

$$\frac{d}{dt}[Y(t)\exp(-\int_{t_0}^{t} k(\tau)d\tau)] \leq 0. \tag{2.133}$$

3. Conclude that $y(t) \leq Y(t) \leq ce^{\int_{t_0}^{t} k(\tau)d\tau}$.

Problem 2.6 (Differentiation Under the Integral Sign) Using the limit definition of the derivative, prove Theorem 2.4, the Leibniz rule for differentiation under the integral sign.

Problem 2.7 (LTV System Representations Determine Linear I/O Systems) Above, we stated two Theorems - 2.6 and 2.8 - which claimed that linear time-varying system representations generate linear I/O dynamical systems. Supply proofs of Theorems 2.6 and 2.8.

Problem 2.8 (An Inverse Initial Value Problem) We know that $\Phi(t, t_0)$ is the solution to the initial value problem $\dot{X}(t) = A(t)X(t),\ X(t_0) = I$. In this problem, we'll find out what $\Phi(t_0, t)$ corresponds to.

1. Consider a continuously differentiable, matrix-valued function $M(\cdot) : \mathbb{R} \to \mathbb{R}^{n\times n}$. Suppose for all $t \in \mathbb{R}$, $M(t)$ is nonsingular. Determine an expression for $\frac{d}{dt}[M^{-1}(t)]$ in terms of $\dot{M}(t)$ and $M^{-1}(t)$.
2. Now, consider a matrix $A(\cdot) \in PC(\mathbb{R}, \mathbb{R}^{n\times n})$. Find an expression for the derivative $\frac{\partial}{\partial\tau}\Phi(t, \tau)$ of the state transition matrix Φ with respect to $A(\cdot)$, in terms of $\Phi(t, t_0)$ and $A(t)$. *You may assume the derivative is being taken at a point where $A(\cdot)$ is continuous.*
3. Prove that $\Phi(t_0, t)$ is the unique solution of the matrix initial value problem,

$$\dot{X}(t) = -X(t)A(t),\ X(t_0) = I. \tag{2.134}$$

Problem 2.9 (The Jacobi-Liouville Formula ★ [8]) Above, we showed that the continuous-time state transition matrix is always invertible. Here, we'll provide another proof of this by means of the *Jacobi-Liouville formula*, which explicitly provides a formula for the determinant of the state transition matrix. In particular, the Jacobi-Liouville formula is,

$$\det \Phi(t, t_0) = \exp\left(\int_{t_0}^{t} \mathrm{tr}(A(\tau))d\tau\right). \tag{2.135}$$

1. Prove that, for $M \in \mathbb{R}^{n\times n}$ and $\epsilon \in \mathbb{R}$, there exists a continuous function $R : \mathbb{R} \to \mathbb{R}$ for which

$$\det(I+\epsilon M) = 1 + \epsilon\,\mathrm{tr}(M) + R(\epsilon) \text{ and } \lim_{\epsilon\to 0}\frac{R(\epsilon)}{\epsilon} = 0. \tag{2.136}$$

Hint: consider working with eigenvalues.

2. Using the determinant formula from (1), show that

$$\frac{d}{dt}\det[\Phi(t,t_0)] = \mathrm{tr}(A(t))\det[\Phi(t,t_0)]. \tag{2.137}$$

Hint: Work with the limit definition of the derivative. If you use a Taylor approximation, be rigorous about your use of the remainder term.

3. Conclude the Jacobi-Liouville formula. Using the Jacobi-Liouville formula, provide a proof that $\Phi(t,t_0)$ is invertible for all $(t,t_0)\in\mathbb{R}\times\mathbb{R}$.

Problem 2.10 (Solution of a Matrix Differential Equation [7]) Let $A_1(\cdot), A_2(\cdot)$, and $F(\cdot)$ be elements of $PC(\mathbb{R},\mathbb{R}^{n\times n})$. Let Φ_i be the state transition matrix of $\dot{x}(t) = A_i(t)x(t)$ for $i=1,2$. Show that the solution of the matrix differential equation:

$$\dot{X}(t) = A_1(t)X(t) + X(t)A_2^\top(t) + F(t),\ X(t_0) = X_0, \tag{2.138}$$

is given by,

$$X(t) = \Phi_1(t,t_0)X_0\Phi_2^\top(t,t_0) + \int_{t_0}^{t}\Phi_1(t,\tau)F(\tau)\Phi_2^\top(t,\tau)d\tau. \tag{2.139}$$

Is this the unique solution of the matrix differential equation? Back up your answer with a proof or disproof.

Problem 2.11 (A Special State Transition Matrix) Consider a piecewise continuous matrix $A\in PC(\mathbb{R},\mathbb{R}^{n\times n})$, and let Φ denote the state transition matrix of $\dot{x}(t) = A(t)x(t)$. If for every $(\tau,t)\in\mathbb{R}\times\mathbb{R}$, one has,

$$A(t)\left(\int_{\tau}^{t}A(\eta)d\eta\right) = \left(\int_{\tau}^{t}A(\eta)d\eta\right)A(t), \tag{2.140}$$

prove using the Peano-Baker series that,

$$\Phi(t,\tau) = \exp\left(\int_{\tau}^{t}A(\eta)d\eta\right) = \sum_{k=0}^{\infty}\frac{1}{k!}\left(\int_{\tau}^{t}A(\eta)d\eta\right)^k. \tag{2.141}$$

Using this result, calculate the state transition matrix associated to the matrix,

$$A(t) = \begin{bmatrix}0 & 0\\ t & 0\end{bmatrix}. \tag{2.142}$$

References

1. Stephen Abbott et al. *Understanding analysis*, volume 2. Springer, 2001.
2. Panos J Antsaklis and Anthony N Michel. *Linear systems*, volume 8. Springer, 1997.
3. Karl Johan Åström and Richard Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.
4. Sheldon Axler. *Measure, integration & real analysis*. Springer Nature, 2020.
5. Sheldon Axler. *Linear algebra done right*. Springer Nature, 2024.
6. Frank M Callier and Charles A Desoer. *Linear system theory*. Springer Science & Business Media, 2012.
7. Chih-Yuan Chiu, Claire Tomlin, and Yi Ma. *Linear Systems*. Available online at https://ucb-ee106.github.io/106b-sp23site/assets/Linear_Systems___Professor_Ma.pdf, 2019.
8. Mohammed Dahleh, Munther A Dahleh, and George Verghese. *Lectures on dynamic systems and control*. Massachusets Institute of Technology, 2004.
9. John C. Doyle. Analysis of feedback systems with structured uncertainties. In *IEE Proceedings D (Control Theory and Applications)*, volume 129, pages 242–250. IET Digital Library, 1982.
10. John C. Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.
11. Geir E Dullerud and Fernando Paganini. *A course in robust control theory: a convex approach*, volume 36. Springer Science & Business Media, 2013.
12. *Linear Dynamical Systems*, 2008.
13. Stephen H Friedberg, Arnold J Insel, and Lawrence E Spence. *Linear Algebra*. Pearson, 2014.
14. Michael Green and David J.N. Limebeer. *Linear Robust Control*. Dover, 1995.
15. Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Dover, 1985.
16. Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
17. Andrew D. Lewis. *A Mathematical Approach to Classical Control*. Online Distribution, 2016.
18. Andrew D. Lewis. *Introduction to Differential Equations*. Online Distribution, 2017.
19. Andrew D. Lewis. *A Mathematical Introduction to Signals and Systems, Volume 4*. Available at https://mast.queensu.ca/ andrew/teaching/SigSys/pdf/volume4.pdf, 2022.
20. John Lygeros and Federico Ramponi. Lecture notes on linear system theory. *Automatic Control Laboratory, ETH Zurich*, 2010.
21. Alexandre Megretski. Multivariable control systems. 2004.
22. Richard M Murray. Feedback systems: Notes on linear systems theory. 2020.
23. Andrew Packard and John Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.
24. Andrew Packard, Roberto Horowitz, Kameshwar Poola, and Francesco Borrelli. *Dynamic Systems and Feedback, Class Notes*. Avaliable online, 2018.
25. Lawrence Perko. *Differential Equations and Dynamical Systems*. Springer, 2000.
26. Ian Postlethwaite and Sigurd Skogestad. *Multivariable Feedback Control, Analysis & Design*. Wiley, 2005.
27. Walter Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987.
28. Wilson J. Rugh. *Linear System Theory*. Prentice Hall, 1996.

29. Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media, 2013.
30. Elias M Stein and Rami Shakarchi. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press, 2009.
31. Elias M Stein and Rami Shakarchi. *Complex analysis*, volume 2. Princeton University Press, 2010.
32. Elias M Stein and Rami Shakarchi. *Fourier analysis: an introduction*, volume 1. Princeton University Press, 2011.
33. Masayoshi Tomizuka. *Advanced Control Systems I.* 2022.
34. Kemin Zhou and John C. Doyle. *Essentials of robust control.* Prentice hall, 1998.
35. Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control.* Prentice hall, 1996.