Lecture Notes for Math 3410, with Computational Examples

Lecture Notes for Math 3410, with Computational Examples

Sean Fitzpatrick

University of Lethbridge

March 22, 2022

Preface

Linear algebra is a mature, rich subject, full of both fascinating theory and useful applications. One of the things you might have taken away from a first course in the subject is that there's a lot of tedious calculation involved. This is true, if you're a human. But the algorithms you learn in a course like Math 1410 are easily implemented on a computer. If we want to be able to discuss any of the interesting applications of linear algebra, we're going to need to learn how to do linear algebra on a computer.

There are many good mathematical software products that can deal with linear algebra, like Maple, Mathematica, and MatLab. But all of these are proprietary, and expensive. Sage is a popular open source system for mathematics, and students considering further studies in mathematics would do well to learn Sage. Since most people in Math 3410 are probably not considering a career as a mathematician, we'll try to do everything in Python.

Python is a very popular programming language, partly because of its ease of use. Those of you enrolled in Education may find yourself teaching Python to your students one day. Also, if you do want to use Sage, you're in luck: Sage is an amalgamation of many different software tools, including Python. So any Python code you encounter in this course can also be run on Sage. You do not have to be a programmer to run the code in this book. We'll be primarily working with the **SymPy** Python library, which provides many easy to use functions for operations like determinant and inverse.

These notes originally began as an attempt to make Python-based worksheets that could be exported from PreTeXt to Jupyter, for use in the classroom. It quickly became apparent that something more was needed, and the worksheets morphed into lecture notes. These are intended to serve as a textbook for Math 3410, but with some work they can also be used in class. The notes are written in PreTeXt, and can be converted to both Jupyter notebooks and reveal.js slides.

I initially wrote these notes during the Fall 2019 semester, for Math 3410 at the University of Lethbridge. The original textbook for the course was Linear Algebra with Applications, by Keith Nicholson. This book is available as an open education resource from Lyryx Learning[1]. Since these notes followed our course, and the course, to some extent, followed Nicholson's book, the notation, terminology, and (to some extent) organization of content aligns with that in Nicholson. (However, in a few places a careful reader might be able to detect the influence of Sheldon Axler's beautiful Linear Algebra Done Right.)

---

[1] lyryx.com/linear-algebra-applications/

Contents

Chapter 1

Vector spaces

In your first course in linear algebra, you likely worked a lot with vectors in two and three dimensions, where they can be visualized geometrically as objects with magnitude and direction (and drawn as arrows). You probably extended your understanding of vectors to include column vectors; that is, $1 \times n$ matrices of the form $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$.

Using either geometric arguments (in $\mathbb{R}^2$ or $\mathbb{R}^3$) or the properties of matrix arithmetic, you would have learned that these vectors can be added, by adding corresponding components, and multiplied by scalars — that is, real numbers — by multiplying each component of the vector by the scalar.

It's also likely, although you may not have spent too long thinking about it, that you looked at the properties obeyed by the addition and scalar multiplication of vectors (or, for that matter, matrices). For example, you may have made use of the fact that order of addition doesn't matter, or that scalar multiplication distributes over addition.

It turns out that the algebraic properties satisfied by vector addition and scalar multiplication are not unique to vectors, as vectors were understood in your first course in linear algebra. In fact, many types of mathematical object exhibit similar behaviour. Examples include matrices, polynomials, and even functions.

Linear algebra, as an abstract mathematical topic, begins with a realization of the importance of these properties. Indeed, these properties, established as theorems for vectors in $\mathbb{R}^n$, become the axioms for the abstract notion of a vector space. The advantage of abstracting these ideas is that any proofs we write that depend only on these axioms will automatically be valid for any set of objects satisfying those axioms. That is, a result that is true for vectors in $\mathbb{R}^2$ is often also true for vectors in $\mathbb{R}^n$, and for matrices, and polynomials, and so on. Mathematicians like to be efficient, and prefer to establish a result once in an abstract setting, knowing that it will then apply to many concrete settings that fit into the framework of the abstract result.

1.1 Abstract vector spaces

Definition 1.1.1

A **real vector space** (or vector space over $\mathbb{R}$) is a nonempty set V, whose objects are called **vectors**, equipped with two operations:

1. **Addition**, which is a map from $V \times V$ to V that associates each ordered pair of vectors $(\mathbf{v}, \mathbf{w})$ to a vector $\mathbf{v} + \mathbf{w}$, called the **sum** of $\mathbf{v}$ and $\mathbf{w}$.

2. **Scalar multiplication**, which is a map from $\mathbb{R} \times V$ to V that associates each real number c and vector $\mathbf{v}$ to a vector $c\mathbf{v}$.

The operations of addition and scalar multiplication are required to satisfy the following axioms:

A1. If $\mathbf{u}, \mathbf{v} \in V$, then $\mathbf{u} + \mathbf{v} \in V$. (Closure under addition)

A2. For all $\mathbf{u}, \mathbf{v} \in V$, $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$. (Commutativity of addition)

A3. For all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$, $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$. (Associativity of addition)

A4. There exists an element $\mathbf{0} \in V$ such that $\mathbf{v} + \mathbf{0} = \mathbf{v}$ for each $\mathbf{v} \in V$. (Existence of a zero vector)

A5. For each $\mathbf{v} \in V$, there exists a vector $-\mathbf{v} \in V$ such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$. (Existence of negatives)

S1. If $\mathbf{v} \in V$, then $c\mathbf{v} \in V$ for all $c \in \mathbb{R}$. (Closure under scalar multiplication)

S2. For all $c \in \mathbb{R}$ and $\mathbf{v}, \mathbf{w} \in V$, $c(\mathbf{v} + \mathbf{w}) = c\mathbf{v} + c\mathbf{w}$. (Distribution over vector addition)

S3. For all $a, b \in \mathbb{R}$ and $\mathbf{v} \in V$, $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$. (Distribution over scalar addition)

S4. For all $a, b \in \mathbb{R}$ and $\mathbf{v} \in V$, $a(b\mathbf{v}) = (ab)\mathbf{v}$. (Associativity of scalar multiplication)

S5. For all $\mathbf{v} \in V$, $1\mathbf{v} = \mathbf{v}$. (Normalization of scalar multiplication)

Note that a zero vector must exist in every vector space. This simple observation is a key component of many proofs and counterexamples in linear algebra. In general, we may define a vector space whose scalars belong to a field $\mathbb{F}$. A field is a set of objects whose algebraic properties are modelled after those of the real numbers. Fields have their [own set of axioms](#)[1], which we will not list here. While it is possible to study linear algebra over finite fields (like the integers modulo a prime number) we will only consider two fields: the real numbers $\mathbb{R}$, and the complex numbers .

A vector space whose scalars are complex numbers will be called a complex vector space. While many students are initially intimidated by the complex numbers, most results in linear algebra work exactly the same over as they do over $\mathbb{R}$. And where the results differ, things are usually easier with complex numbers, owing in part to the fact that all complex polynomials can be completely factored.

To help us gain familiarity with the abstract nature of Definition **??**, let us consider

---

[1] en.wikipedia.org/wiki/Field_(mathematics)#Classic_definition

some basic examples.

> **Example 1.1.2**
>
> *The following are examples of vector spaces. We leave verification of axioms as an exercise.*
>
> 1. *The set $\mathbb{R}^n$ of n-tuples $(x_1, x_2, \ldots, x_n)$ of real numbers, where we define*
>
> $$(x_1, x_2, \ldots, x_n) + (y_1, y_2, \ldots, y_n) = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$
> $$c(x_1, x_2, \ldots, x_n) = (cx_1, cx_2, \ldots, cx_n).$$
>
>   *We will also often use $\mathbb{R}^n$ to refer to the vector space of $1 \times n$ column matrices $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, where addition and scalar multiplication are defined as for matrices (and the same as the above, with the only difference being the way in which we choose to write our vectors). If the distinction between n-tuples and column matrices is ever important, it will be made clear.*
>
> 2. *The set $\mathbf{M}_{mn}$ of $m \times n$ matrices, equipped with the usual matrix addition and scalar multiplication.*
>
> 3. *The set $\mathbf{P}_n$ of all polynomials*
>
> $$p(x) = a_0 + a_1 x + \cdots + a_n x^n$$
>
>   *of degree less than or equal to n, where, for*
>
> $$p(x) = a_0 + a_1 x + \cdots + a_n x^n$$
> $$q(x) = b_0 + b_1 x + \cdots + b_n x^n$$
>
>   *we define*
>
> $$p(x) + q(x) = (a_0 + b_0) + (a_1 + b_1)x + \cdots + (a_n + b_n)x^n$$
>
>   *and*
>
> $$cp(x) = ca_0 + (ca_1)x + \cdots + (ca_n)x^n.$$
>
>   *The zero vector is the polynomial $0 = 0 + 0x + \cdots + 0x^n$.*
>
> 4. *The set $\mathbf{F}[a, b]$ of all functions $f : [a, b] \to \mathbb{R}$, where we define $(f + g)(x) = f(x) + g(x)$ and $(cf)(x) = c(f(x))$. The zero function is the function satisfying $0(x) = 0$ for all $x \in [a, b]$, and the negative of a function f is given by $(-f)(x) = -f(x)$ for all $x \in [a, b]$.*

Other common examples of vector spaces can be found online; for example, on Wikipedia[2]. It is also interesting to try to think of less common examples. For example, can you think of a way to define a vector space structure on the set $(0, \infty)$ of all positive real numbers?

There are a number of other algebraic properties that are common to all vector spaces; for example, it is true that $0\mathbf{v} = \mathbf{0}$ for all vectors $\mathbf{v}$ in any vector space V.

---

[2] en.wikipedia.org/wiki/Examples_of_vector_spaces

The reason these are not included is that the ten axioms in Definition ?? are the ones deemed "essential" – all other properties can be deduced from the axioms. To demonstrate, we next give the proof that $0\mathbf{v} = \mathbf{0}$.

> **Theorem 1.1.3**
>
> **In any vector space V, we have $0\mathbf{v} = \mathbf{0}$ for all $\mathbf{v} \in V$**

Proof. *Since $0 + 0 = 0$, we have $0\mathbf{v} = (0 + 0)\mathbf{v}$. Using the distributive axiom S3, this becomes*

$$0\mathbf{v} + 0\mathbf{v} = 0\mathbf{v}.$$

By axiom A5, there is an element $-0\mathbf{v} \in V$ such that $0\mathbf{v} + (-0\mathbf{v}) = \mathbf{0}$. Adding this to both sides of the equation above, we get:

$$(0\mathbf{v} + 0\mathbf{v}) + (-0\mathbf{v}) = 0\mathbf{v} + (-0\mathbf{v}).$$

Now, apply the associative property (A3) on the left, and A5 on the right, to get

$$0\mathbf{v} + (0\mathbf{v} + (-0\mathbf{v})) = \mathbf{0}.$$

Using A5 again on the left, we get $0\mathbf{v} + \mathbf{0} = \mathbf{0}$. Finally, axiom A4 guarantees $0\mathbf{v} = 0\mathbf{v} + \mathbf{0} = \mathbf{0}$. ∎

Similar tactics can be used to establish the following results, which we leave as an exercise. Solutions are included, but it will be worth your while in the long run to wrestle with these.

> **Exercise 1.1.4**
>
> *Show that the following properties are valid in any vector space V:*
>
> 1. *If $\mathbf{u} + \mathbf{v} = \mathbf{u} + \mathbf{w}$, then $\mathbf{v} = \mathbf{w}$.*
>
> 2. *For any scalar c, $c\mathbf{0} = \mathbf{0}$.*
>
> 3. *If $c\mathbf{v} = \mathbf{0}$, then either $c = 0$ or $\mathbf{v} = \mathbf{0}$.*
>
> 4. *For any $\mathbf{v} \in V$, $(-1)\mathbf{v} = -\mathbf{v}$.*
>
> 5. *The zero vector is the unique vector such that $\mathbf{v} + \mathbf{0} = \mathbf{v}$ for all $\mathbf{v} \in V$.*
>
> 6. *The negative $-\mathbf{v}$ of any vector $\mathbf{v}$ is unique.*

1.2 Subspaces

We begin with a motivating example. Let $\mathbf{v}$ be a nonzero vector in some vector space V. Consider the set $S = \{c\mathbf{v} \mid c \in \mathbb{R}\}$. Given $a\mathbf{v}, b\mathbf{v} \in S$, notice that $a\mathbf{v} + b\mathbf{v} = (a + b)\mathbf{v}$ is also an element of S, since $a + b$ is again a real number. Moreover, for any real number c, $c(a\mathbf{v}) = (ca)\mathbf{v}$ is an element of S.

There are two important observations: one is that performing addition or scalar multiplication on elements of S produces a new element of S. The other is that this addition and multiplication is essentially that of $\mathbb{R}$. The vector $\mathbf{v}$ is just a placeholder. Addition simply involves the real number addition $a+b$. Scalar multiplication becomes the real number multiplication ca. So we expect that the rules for addition and scalar multiplication in S follow those in $\mathbb{R}$, so that S is like a "copy" of $\mathbb{R}$ inside of V. In particular, addition and scalar multiplication in S will satisfy all the vector space axioms,

so that S deserves to be considered a vector space in its own right.

A similar thing happens if we consider a set $U = \{a\mathbf{v} + b\mathbf{w} \mid a, b \in \mathbb{R}\}$, where $\mathbf{v}, \mathbf{w}$ are two vectors in a vector space V. Given two elements $a_1\mathbf{v} + a_2\mathbf{w}, b_1\mathbf{u} + b_2\mathbf{w}$, we have

$$(a_1\mathbf{v} + a_2\mathbf{w}) + (b_1\mathbf{v} + b_2\mathbf{w}) = (a_1 + b_1)\mathbf{v} + (a_2 + b_2)\mathbf{w},$$

which is again an element of U, and the addition rule looks an awful lot like the addition rule $(a_1, a_2) + (b_1, b_2) = (a_1 + b_1, a_2 + b_2)$ in $\mathbb{R}^2$. Scalar multiplication follows a similar pattern.

In general we are often interested in subsets of vectors spaces that behave like "copies" of smaller vector spaces contained within the larger space. The technical term for this is subspace.

Definition 1.2.1

*Let V be a vector space, and let $U \subseteq V$ be a subset. We say that U is a **subspace** of V if U is itself a vector space when using the addition and scalar multiplication of V.*

If we were to follow the definition, then verifying that a subset U is a subspace would involve checking all ten vector space axioms. Fortunately, this is not necessary. Since the operations are those of the vector space V, most properties follow automatically, being inherited from those of V.

Theorem 1.2.2

Let V be a vector space and let $U \subseteq V$ be a subset. Then U is a subspace of V if and only if the following conditions are satisfied:

1. $\mathbf{0} \in U$, where $\mathbf{0}$ is the zero vector of V.

2. U is closed under addition. That is, for all $\mathbf{u}_1, \mathbf{u}_2 \in U$, we have $\mathbf{u}_1 + \mathbf{u}_2 \in U$.

3. U is closed under scalar multiplication. That is, for all $\mathbf{u} \in U$ and $c \in \mathbb{R}$, $c\mathbf{u} \in U$.

Proof. *If U is a vector space, then clearly the second and third conditions must hold. Since a vector space must be nonempty, there is some $\mathbf{u} \in U$, from which it follows that $\mathbf{0} = 0\mathbf{u} \in U$.*

Conversely, if all three conditions hold, we have axioms A1, A4, and S1 by assumption. Axioms A2 and A3 hold since any vector in U is also a vector in V; the same reasoning shows that axioms S2, S3, S4, and S5 hold. Finally, axiom A5 holds because condition 3 ensures that $(-1)\mathbf{u} \in U$ for any $\mathbf{u} \in U$, and we know that $(-1)\mathbf{u} = -\mathbf{u}$ by Exercise ??. ∎

In some texts, the condition that $\mathbf{0} \in U$ is replaced by the requirement that U be nonempty. Existence of $\mathbf{0}$ then follows from the fact that $0\mathbf{v} = \mathbf{0}$. However, it is usually easy to check that a set contains the zero vector, so it's the first thing one typically looks for when confirming that a subset is nonempty.

Example 1.2.3

For any vector space V, the set $\{\mathbf{0}\}$ is a subspace, known as the trivial subspace.

If $V = \mathbf{P}$ is the vector space of all polynomials, then for any natural number n, the subset U of all polynomials of degree less than or equal to n is a sub-

space of V. Another common type of polynomial subspace is the set of all poly-
nomials with a given root. For example, the set $U = \{p(x) \in \textbf{P} \,|\, p(1) = 0\}$
is easily confirmed to be a subspace. However, a condition such as $p(1) = 2$
would not define a subspace, since this condition is not satisfied by the zero
polynomial.

In $\mathbb{R}^n$, we can define a subspace using one or more homogeneous linear
equations. For example, the set

$$\{(x, y, z) \,|\, 2x - 3y + 4z = 0\}$$

is a subspace of $\mathbb{R}^3$. A non-homogeneous equation won't work, however, since
it would exclude the zero vector. Of course, we should expect that any non-
linear equation fails to define a subspace, although one is still expected to
verify this by confirming the failure of one of the axioms. For example, the set
$S = \{(x, y) \,|\, x = y^2\}$ is not a subspace; although it contains the zero vector
(since $0^2 = 0$), we have $(1, 1) \in S$, but $2(1, 1) = (2, 2)$ does not belong to S.

In the next section, we'll encounter perhaps the most fruitful source of subspaces:
sets of linear combinations (or spans).

1.3 Span

*Recall that a **linear combination** of a set of vectors $v_1, \ldots, v_k$ is a vector expression*
of the form

$$w = c_1 v_1 + c_2 v_2 + \cdots + c_k v_k,$$

where $c_1, \ldots, c_k$ are scalars.

*The **span** of those same vectors is the set of all possible linear combinations:*

$$span\{v_1, \ldots, v_k\} = \{c_1 v_1 + \cdots + c_k v_k \,|\, c_1, \ldots, c_k \in \mathbb{F}\}.$$

Therefore, the questions "Is the vector w in $span\{v_1, \ldots, v_k\}$?" is really asking, "Can
w be written as a linear combination of $v_1, \ldots, v_k$?"

With the appropriate setup, all such questions become questions about solving
systems of equations. Here, we will look at a few such examples.

Exercise 1.3.1

Determine whether the vector $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ is in the span of the vectors $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\begin{bmatrix} -1 \\ 2 \end{bmatrix}$.

Our next example involves polynomials. At first this looks like a different problem,
but it's essentially the same once we set it up.

Exercise 1.3.2

Determine whether $p(x) = 1 + x + 4x^2$ belongs to $span\{1 + 2x - x^2, 3 + 5x + 2x^2\}$.

One of the reasons we care about linear combinations and span is that it gives us
an easy means of generating subspaces, as the following theorem suggests.

> ### Theorem 1.3.3
>
> Let V be a vector space, and let $v_1, v_2, \ldots, v_k$ be vectors in V. Then:
>
> 1. $U = span\{v_1, v_2, \ldots, v_k\}$ is a subspace of V.
>
> 2. U is the smallest subspace of V containing $v_1, \ldots, v_k$, in the sense that if $W \subseteq V$ is a subspace and $v_1, \ldots, v_k \in W$, then $U \subseteq W$.

Proof. Let $U = span\{v_1, v_2, \ldots, v_k\}$. Then $0 \in U$, since $0 = 0v_1 + 0v_2 + \cdots + 0v_k$. If $u = a_1v_1 + a_2v_2 + \cdots + a_kv_k$ and $w = b_1v_1 + b_2v_2 + \cdots + b_kv_k$ are vectors in U, then

$$u + w = (a_1v_1 + a_2v_2 + \cdots + a_kv_k) + (b_1v_1 + b_2v_2 + \cdots + b_kv_k)$$
$$= (a_1 + b_1)v_1 + (a_2 + b_2)v_2 + \cdots + (a_k + b_k)v_k$$

is in U, and

$$cu = c(a_1v_1 + a_2v_2 + \cdots + a_kv_k)$$
$$= (ca_1)v_1 + (ca_2)v_2 + \cdots + (ca_k)v_k$$

is in U, so by Theorem ??, U is a subspace.

To see that U is the smallest subspace containing $v_1, \ldots, v_k$, we need only note that if $v_1, \ldots, v_k \in W$, where W is a subspace, then since W is closed under scalar multiplication, we know that $c_1v_1, \ldots, c_kv_k$ for any scalars $c_1, \ldots, c_k$, and since W is closed under addition, $c_1v_1 + \cdots + c_kv_k \in W$. Thus, W contains all linear combinations of $v_1, \ldots, v_k$, which is to say that W contains U. ■

We end this section with a few non-computational, but useful results, which will be left as exercises to be done in class, or by the reader.

> ### Exercise 1.3.4
>
> Let V be a vector space, and let $X, Y \subseteq V$. Show that if $X \subseteq Y$, then span $X \subseteq$ span Y.

> ### Exercise 1.3.5
>
> Can $\{(1, 2, 0), (1, 1, 1)\}$ span $\{(a, b, 0) \mid a, b \in \mathbb{R}\}$?

> ### Theorem 1.3.6
>
> Let V be a vector space, and let $v_1, \ldots, v_k \in V$. If $u \in span\{v_1, \ldots, v_k\}$, then
> $$span\{u, v_1, \ldots, v_k\} = span\{v_1, \ldots, v_k\}.$$

*The moral of Theorem ?? is that one vector in a set is a linear combination of the others, we can remove it from the set without affecting the span. This suggests that we might want to look for the most "efficient" spanning sets – those in which no vector in the set can be written in terms of the others. Such sets are called **linearly independent**, and they are the subject of Section ??.*

1.4 Worksheet: understanding span

In this worksheet, we will attempt to understand the concept of span. Recall from Section **??** that the span of a set of vectors $v_1, v_2, \ldots, v_k$ in a vector space V is the set of all linear combinations that can be generated from those vectors.

Therefore, the question "Does the vector w belong to the span of $v_1, v_2, \ldots, v_k$?" is equivalent to asking, "Can I write w as a linear combination of the v_i?", which, in turn, is equivalent to asking:

Do there exist scalars $c_1, c_2, \ldots, c_k$ such that

$$w = c_1 v_1 + c_2 v_2 + \cdots + c_k v_k ?$$

In any finite-dimensional vector space, this last question can be turned into a system of equations. If that system has a solution, then yes — your vector is in the span. If the system is inconsistent, then the answer is no.

1.   Determine whether or not the vector $w = \langle 3, -1, 4, 2 \rangle$ in $\mathbb{R}^4$ belongs to the span of the vectors

$$\langle 2, 1, 4, -3 \rangle, \langle 0, 2, 1, 4 \rangle, \langle -1, 1, 0, 2 \rangle.$$

To assist with solving this problem, a code cell is provided below. Once you have determined the augmented matrix of your system of equations, see Section **??** for details on how to enter your matrix, and then compute its reduced row-echelon form.

```
from sympy import Matrix, init_printing
init_printing()
```

2.   Determine whether or not the polynomial $q(x) = 4 - 6x - 11x^2$ belongs to the span of the polynomials

$$p_1(x) = x - 3x^2, p_2(x) = 2 - x, p_3(x) = -1 + 4x + x^2.$$

For our next activity, we are going to look at RGB colours. Here, RGB stands for Red, Green, Blue. All colours displayed by your computer monitor can be expressed in terms of these colours.

First, we load some Python libraries we'll need. These are intended for use in a Jupyter notebook and won't run properly if you are using Sagecell in the HTML textbook.

```
import ipywidgets as wid
import matplotlib.pyplot as plt
```

Next, we will create a widget that lets us select values for red, green, and blue. The RGB colour system assigns 8-bit values to each colour. Possible values for each range from 0 to 255; this indicates how much of each colour will be blended to create the colour you want. Extensive information on the RGB colour system can be found on wikipedia[1], and there are a number of good online resources about the use of RGB in web design, such as this one from w3schools[2].

```
r=wid.IntSlider(
    value=155,
    min=0,
    max=255,
    step=1,
    description='Red:'
)
g=wid.IntSlider(
    value=155,
    min=0,
    max=255,
    step=1,
    description='Green:'
)
b=wid.IntSlider(
    value=155,
    min=0,
    max=255,
    step=1,
    description='Blue:'
)
display(r,g,b)
```

By moving the sliders generated above, you can create different colours. To see what colour you've created by moving the sliders, run the code below.

```
plt.imshow([[(r.value/255, g.value/255, b.value/255)]])
```

3. In what ways can you explain the RGB colour system in terms of span?

4. Why would it nonetheless be inappropriate to describe the set of all RGB colours as a vector space?

---

[1]en.wikipedia.org/wiki/RGB_color_model
[2]www.w3schools.com/colors/colors_rgb.asp

1.5 Linear Independence

In any vector space V, we say that a set of vectors

$$\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$$

is **linearly independent** if for any scalars $c_1, \ldots, c_k$

$$c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k = \mathbf{0} \quad \Rightarrow \quad c_1 = \cdots = c_k = 0.$$

This means that no vector in the set can be written as a linear combination of the other vectors in that set. We will soon see that when looking for vectors that span a subspace, it is especially useful to find a spanning set that is also linearly independent. The following lemma establishes some basic properties of independent sets.

> **Lemma 1.5.1**
>
> In any vector space V:
>
> 1. If $\mathbf{v} \neq \mathbf{0}$, then $\{\mathbf{v}\}$ is independent.
>
> 2. If $S \subseteq V$ contains the zero vector, then S is dependent.

The definition of linear independence tells us that if $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ is an independent set of vectors, then there is only one way to write $\mathbf{0}$ as a linear combination of these vectors; namely,

$$\mathbf{0} = 0\mathbf{v}_1 + 0\mathbf{v}_2 + \cdots + 0\mathbf{v}_k.$$

In fact, more is true: any vector in the span of a linearly independent set can be written in only one way as a linear combination of those vectors.

Computationally, questions about linear independence are just questions about homogeneous systems of linear equations. For example, suppose we want to know if the vectors

$$\mathbf{u} = \begin{bmatrix} 1 \\ -1 \\ 4 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} 0 \\ 2 \\ -3 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 4 \\ 0 \\ -3 \end{bmatrix}$$

are linearly independent in $\mathbb{R}^3$. This question leads to the vector equation

$$x\mathbf{u} + y\mathbf{v} + z\mathbf{w} = \mathbf{0},$$

which becomes the matrix equation

$$\begin{bmatrix} 1 & 0 & 4 \\ -1 & 2 & 0 \\ 4 & -3 & -3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

We now apply some basic theory from linear algebra. A unique (and therefore, trivial) solution to this system is guaranteed if the matrix $A = \begin{bmatrix} 1 & 0 & 4 \\ -1 & 2 & 0 \\ 4 & -3 & -3 \end{bmatrix}$ is invertible, since in that case we have $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A^{-1}\mathbf{0} = \mathbf{0}$.

This approach is problematic, however, since it won't work if we have 2 vectors, or 4. Instead, we look at the reduced row-echelon form. A unique solution corresponds to having a leading 1 in each column of A. Let's check this condition.

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix(3,3,[1,0,4,-1,2,0,4,-3,-3])
A.rref()
```

$$\bbm 1\amp 0\amp 0\\0\amp 1\amp 0\\0\amp 0\amp 1\ebm,$$
$$(0,1,2)$$

One observation is useful here, and will lead to a better understanding of indepen-
dence. First, it would be impossible to have 4 or more linearly independent vectors in
$\mathbb{R}^3$. Why? (How many leading ones can you have in a 3×4 matrix?) Second, having
two or fewer vectors makes it more likely that the set is independent.

The largest set of linearly independent vectors possible in $\mathbb{R}^3$ contains three vec-
tors. You might have also observed that the smallest number of vectors needed to
span $\mathbb{R}^3$ is 3. Hmm. Seems like there's something interesting going on here. But first,
some more computation.

Exercise 1.5.2

Determine whether the set $\left\{ \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ 4 \\ 9 \end{bmatrix} \right\}$ is linearly independent
in $\mathbb{R}^3$.

Exercise 1.5.3

Which of the following subsets of $P_2(\mathbb{R})$ are independent?

$$\text{(a) } S_1 = \{x^2 + 1, x + 1, x\}$$
$$\text{(b) } S_2 = \{x^2 - x + 3, 2x^2 + x + 5, x^2 + 5x + 1\}$$

Exercise 1.5.4

Determine whether or not the set

$$\left\{ \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \right\}$$

is linearly independent in $M_2(\mathbb{R})$.

1.6 Basis and dimension

Next, we begin with an important result, sometimes known as the "Fundamental The-
orem":

Theorem 1.6.1 Fundamental Theorem (Steinitz Exchange Lemma).

Suppose $V = span\{v_1, \ldots, v_n\}$. If $\{w_1, \ldots, w_m\}$ is a linearly indepen-
dent set of vectors in V, then $m \leq n$.

If a set of vectors spans a vector space V, and it is not independent, we observed
that it is possible to remove a vector from the set and still span V. This suggests that

spanning sets that are also linearly independent are of particular importance, and indeed, they are important enough to have a name.

Definition 1.6.2

*Let V be a vector space. A set $\mathcal{B} = \{e_1, \ldots, e_n\}$ is called a **basis** of V if $\mathcal{B}$ is linearly independent, and span $\mathcal{B} = V$.*

The importance of a basis is that vector vector $v \in V$ can be written in terms of the basis, and this expression as a linear combination of basis vectors is unique. Another important fact is that every basis has the same number of elements.

Theorem 1.6.3 Invariance Theorem.

If $\{e_1, \ldots, e_n\}$ and $\{f_1, \ldots, f_m\}$ are both bases of a vector space V, then $m = n$.

Suppose $V = \text{span}\{v_1, \ldots, v_n\}$. If this set is not linearly independent, Theorem ?? tells us that we can remove a vector from the set, and still span V. We can repeat this procedure until we have a linearly independent set of vectors, which will then be a basis. These results let us make a definition.

Definition 1.6.4

*Let V be a vector space. If V can be spanned by a finite number of vectors, then we call V a **finite-dimensional** vector space. If V is finite-dimensional (and non-trivial), and $\{e_1, \ldots, e_n\}$ is a basis of V, we say that V has **dimension** n, and write*

$$\dim V = n.$$

*If V cannot be spanned by finitely many vectors, we say that V is **infinite-dimensional**.*

Exercise 1.6.5

Find a basis for $U = \{X \in M_{22} \mid XA = AX\}$, if $A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$

Example 1.6.6 Standard bases.

Most of the vector spaces we work with come equipped with a standard basis. The standard basis for a vector space is typically a basis such that the scalars needed to express a vector in terms of that basis are the same scalars used to define the vector in the first place. For example, we write an element of $\mathbb{R}^3$ as

(x, y, z) *(or $\langle x, y, z \rangle$, or $\begin{bmatrix} x & y & z \end{bmatrix}$, or $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ ...). We can also write*

$$(x, y, z) = x(1, 0, 0) + y(0, 1, 0) + z(0, 0, 1).$$

The set $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ is the standard basis for $\mathbb{R}^3$. In general,

the vector space $\mathbb{R}^n$ (written this time as column vectors) has standard basis

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, e_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

From this, we can conclude (unsurprisingly) that dim $\mathbb{R}^n = n$.

Similarly, a polynomial $p(x) \in P_n(\mathbb{R})$ is usually written as

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

suggesting the standard basis $\{1, x, x^2, \ldots, x^n\}$. As a result, we see that dim $P_n(\mathbb{R}) = n + 1$.

For one more example, we note that a 2×2 matrix $A \in M_{22}(\mathbb{R})$ can be written as

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = a \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + c \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + d \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

which suggests a standard basis for $M_{22}(\mathbb{R})$, with similar results for any other matrix space. From this, we can conclude (exercise) that dim $M_{mn}(\mathbb{R}) = mn$.

Solution.

Exercise 1.6.7

Show that the following sets are bases of $\mathbb{R}^3$.

  (a) $\{(1, 1, 0), (1, 0, 1), (0, 1, 1)\}$

  (b) $\{(-1, 1, 1), (1, -1, 1), (1, 1, -1)\}$

The next two exercises are left to the reader to solve. In each case, your goal should be to turn the questions of independence and span into a system of equations, which you can then solve using the computer.

Exercise 1.6.8

Show that the following is a basis of M_{22}:

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \right\}.$$

Exercise 1.6.9

Show that $\{1 + x, x + x^2, x^2 + x^3, x^3\}$ is a basis for P_3.

Exercise 1.6.10

Find a basis and dimension for the following subspaces of P_2:

  a $U_1 = \{a(1 + x) + b(x + x^2) \mid a, b \in \mathbb{R}\}$

  b $U_2 = \{p(x) \in P_2 \mid p(1) = 0\}$

$\quad$ c $\ U_3 = \{p(x) \in P_2 \,|\, p(x) = p(-x)\}$

We've noted a few times now that if $w \in span\{v_1, \ldots, v_n\}$, then

$$span\{w, v_1, \ldots, v_n\} = span\{v_1, \ldots, v_n\}$$

If w is not in the span, we can make another useful observation:

> **Lemma 1.6.11 Independent Lemma.**
>
> **Suppose $\{v_1, \ldots, v_n\}$ is a linearly independent set of vectors in a vector space V. If $u \in V$ but $u \notin span\{v_1, \ldots, v_n\}$, then $\{u, v_1, \ldots, v_n\}$ is independent.**

Proof. *Suppose $S = \{v_1, \ldots, v_n\}$ is independent, and that $u \notin span\, S$. Suppose we have*

$$a u + c_1 v_1 + c_2 v_2 + \cdots + c_n b_n = 0$$

for scalars $a, c_1, \ldots, c_n$. We must have $a = 0$; otherwise, we can multiply by $\frac{1}{a}$ and rearrange to obtain

$$u = -\frac{c_1}{a} v_1 - \cdots - \frac{c_n}{a} v_n,$$

but this would mean that $u \in span\, S$, contradicting our assumption.
$\quad$ *With $a = 0$ we're left with*

$$c_1 v_1 + c_2 v_2 + \cdots + c_n b_n = 0,$$

and since we assumed that the set S is independent, we must have $c_1 = c_2 = \cdots = c_n = 0$. Since we already showed $a = 0$, this shows that $\{u, v_1, \ldots, v_n\}$ is independent. ∎
$\quad$ *This is, in fact, an "if and only if" result. If $u \in span\{v_1, \ldots, v_n\}$, then $\{u, v_1, \ldots, v_n\}$ is not independent. Above, we argued that if V is finite dimensional, then any spanning set for V can be reduced to a basis. It probably won't surprise you that the following is also true.*

> **Lemma 1.6.12**
>
> **Let V be a finite-dimensional vector space, and let U be any subspace of V. Then any independent set of vectors $\{u_1, \ldots, u_k\}$ in U can be enlarged to a basis of U.**

Proof. *This follows from Lemma **??**. If our independent set of vectors spans U, then it's a basis and we're done. If not, we can find some vector not in the span, and add it to our set to obtain a larger set that is still independent. We can continue adding vectors in this fashion until we obtain a spanning set.*
$\quad$ *Note that this process must terminate: V is finite-dimensional, so there is a finite spanning set for V, and therefore for U. By the Steinitz Exchange lemma, our independent set cannot get larger than this spanning set.* ∎

> **Theorem 1.6.13**
>
> **Any finite-dimensional vector space V has a basis. Moreover:**
>
> 1. **If V can be spanned by m vectors, then $dim\, V \leq m$.**
>
> 2. **Given an independent set I in V, and a basis $\mathcal{B}$ of V, we can enlarge I to a basis of V by adding elements of $\mathcal{B}$.**

> If U is a subspace of V, then:
>
> 1. U is finite-dimensional, and $dim\, U \leq dim\, V$.
>
> 2. If $dim\, U = dim\, V$, then $U = V$.

Exercise 1.6.14

Find a basis of $M_{22}(\mathbb{R})$ that contains the vectors

$$\mathbf{v} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

Exercise 1.6.15

Extend the set $\{1 + x, x + x^2, x - x^3\}$ to a basis of $P_3(\mathbb{R})$.

Exercise 1.6.16

Give two examples of infinite-dimensional vector spaces. Support your answer.

Let's recap our results so far:

- *A basis for a vector space V is an independent set of vectors that spans V.*

- *The number of vectors in any basis of V is a constant, called the dimension of V.*

- *The number of vectors in any independent set is always less than or equal to the number of vectors in a spanning set.*

- *In a finite-dimensional vector space, any independent set can be enlarged to a basis, and any spanning set can be cut down to a basis by deleting vectors that are in the span of the remaining vectors.*

Another important aspect of dimension is that it reduces many problems, such as determining equality of subspaces, to counting problems.

Theorem 1.6.17

Let U and W be subspaces of a finite-dimensional vector space V.

1. If $U \subseteq W$, then $dim\, U \leq dim\, W$.

2. If $U \subseteq W$ and $dim\, U = dim\, W$, then $U = W$.

Proof.

1. Suppose $U \subseteq W$, and let $B = \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ be a basis for U. Since B is a basis, it's independent. And since $B \subseteq U$ and $U \subseteq W$, $B \subseteq W$. Thus, B is an independent subset of W, and since any basis of W spans W, we know that $dim\, U = k \leq dim\, W$, by Theorem **??**.

2. Suppose $U \subseteq W$ and $dim\, U = dim\, W$. Let B be a basis for U. As above, B is an independent subset of W. If $W \neq U$, then there is some $\mathbf{w} \in W$ with $\mathbf{w} \notin U$. But $U = span\, B$, so that would mean that $B \cup \{\mathbf{w}\}$ is a linearly independent set containing $dim\, U + 1$ vectors. This is impossible, since $dim\, W = dim\, U$, so

no independent set can contain more than dim U vectors.

∎

An even more useful counting result is the following:

Theorem 1.6.18

Let V be an n-**dimensional vector space. If the set** S **contains** n **vectors, then** S **is independent if and only if** *span* $S = V$.

Proof. *If S is independent, then it can be extended to a basis B with $S \subseteq B$. But S and B both contain n vectors (since dim $V = n$), so we must have $S = B$.*

If S spans V, then S must contain a basis B, and as above, since S and B contain the same number of vectors, they must be equal. ∎

New subspaces from old. *On your first assignment, you showed that if U and W are subspaces of a vector space V, then the intersection $U \cap W$ is also a subspace of V. You also showed that the union $U \cup W$ is generally not a subspace, unless one subspace is contained in the other (in which case the union is just the larger of the two subspaces we already have).*

*In class, we discussed the fact that the right way to define a subspace containing both U and W is using their **sum**: we define the sum $U + W$ of two subspaces by*

$$U + W = \{\boldsymbol{u} + \boldsymbol{w} \,|\, \boldsymbol{u} \in U \text{ and } \boldsymbol{w} \in W\}.$$

We proved that $U + W$ is again a subspace of V.

*If $U \cap W = \{\boldsymbol{0}\}$, we say that the sum is a **direct sum**, and write it as $U \oplus W$. The following theorem might help us understand why direct sums are singled out for special attention:*

Theorem 1.6.19

Let U and W **be subspaces of a finite-dimensional vector space** V. **Then** $U + W$ **is finite-dimensional, and**

$$dim(U + W) = dim\, U + dim\, W - dim(U \cap W).$$

If the sum is direct, then we have simply $dim(U \oplus W) = dim\, U + dim\, W$. The other reason why direct sums are preferable, is that any $\boldsymbol{v} \in U \oplus W$ can be written uniquely as $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}$ where $\boldsymbol{u} \in U$ and $\boldsymbol{w} \in W$.

Theorem 1.6.20

For any subspaces U, W **of a vector space** V, $U \cap W = \{\boldsymbol{0}\}$ **if and only if for every** $\boldsymbol{v} \in U + W$ **there exist unique** $\boldsymbol{u} \in U, \boldsymbol{w} \in W$ **such that** $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}.$

Proof. *Suppose that $U \cap W = \{\boldsymbol{0}\}$, and suppose that we have $\boldsymbol{v} = \boldsymbol{u}_1 + \boldsymbol{w}_1 = \boldsymbol{u}_2 + \boldsymbol{w}_2$, for $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U, \boldsymbol{w}_1, \boldsymbol{w}_2 \in W$. Then $\boldsymbol{0} = (\boldsymbol{u}_1 - \boldsymbol{u}_2) + (\boldsymbol{w}_1 - \boldsymbol{w}_2)$, which implies that*

$$\boldsymbol{w}_1 - \boldsymbol{w}_2 = -(\boldsymbol{u}_1 - \boldsymbol{u}_2).$$

Now, $\boldsymbol{u} = \boldsymbol{u}_1 - \boldsymbol{u}_2 \in U$, since U is a subspace, and similarly, $\boldsymbol{w} = \boldsymbol{w}_1 - \boldsymbol{w}_2 \in W$. But we also have $\boldsymbol{w} = -\boldsymbol{u}$, which implies that $\boldsymbol{w} \in U$. Therefore, $\boldsymbol{w} \in U \cap W$, which implies that $\boldsymbol{w} = \boldsymbol{0}$, so $\boldsymbol{w}_1 = \boldsymbol{w}_2$. But we must also then have $\boldsymbol{u} = \boldsymbol{0}$, so $\boldsymbol{u}_1 = \boldsymbol{u}_2$.

Conversely, suppose that every $\boldsymbol{v} \in U + W$ can be written uniquely as $\boldsymbol{v} = \vec{u} + \boldsymbol{w}$, with $\boldsymbol{u} \in U$ and $\boldsymbol{w} \in W$. Suppose that $\boldsymbol{a} \in U \cap W$. Then $\boldsymbol{a} \in U$ and $\boldsymbol{a} \in W$, so we also have $-\boldsymbol{a} \in W$, since W is a subspace. But then $\boldsymbol{0} = \boldsymbol{a} + (-\boldsymbol{a})$, where $\boldsymbol{a} \in U$ and

$-\boldsymbol{a} \in W$. On the other hand, $\boldsymbol{0} = \boldsymbol{0} + \boldsymbol{0}$, and $\boldsymbol{0}$ belongs to both U and W. It follows that $\boldsymbol{a} = \boldsymbol{0}$. Since $\boldsymbol{a}$ was arbitrary, $U \cap W = \{\boldsymbol{0}\}$. ■

We end with one last application of the theory we've developed on the existence of a basis for a finite-dimensional vector space. As we continue on to later topics, we'll find that it is often useful to be able to decompose a vector space into a direct sum of subspaces. Using bases, we can show that this is always possible.

> **Theorem 1.6.21**
>
> **Let V be a finite-dimensional vector space, and let U be any subspace of V. Then there exists a subspace $W \subseteq V$ such that $U \oplus W = V$.**

Proof. Let $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$ be a basis of U. Since $U \subseteq V$, the set $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$ is a linearly independent subset of V. Since any linearly independent set can be extended to a basis of V, there exist vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$ such that

$$\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$$

is a basis of V.

Now, let $W = \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$. Then W is a subspace, and $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$ is a basis for W. (It spans, and must be independent since it's a subset of an independent set.)

Clearly, $U + W = V$, since $U + W$ contains the basis for V we've constructed. To show the sum is direct, it suffices to show that $U \cap W = \{\boldsymbol{0}\}$. To that end, suppose that $\boldsymbol{v} \in U \cap W$. Since $\boldsymbol{v} \in U$, we have

$$\boldsymbol{v} = a_1 \boldsymbol{u}_1 + \cdots + a_m \boldsymbol{u}_m$$

for scalars $a_1, \ldots, a_m$. Since $\boldsymbol{v} \in W$, we can write

$$\boldsymbol{v} = b_1 \boldsymbol{w}_1 + \cdots + b_n \boldsymbol{w}_n$$

for scalars $b_1, \ldots, b_n$. But then

$$\boldsymbol{0} = \boldsymbol{v} - \boldsymbol{v} = a_1 \boldsymbol{u}_1 + \cdots a_m \boldsymbol{u}_m - b_1 \boldsymbol{w}_1 - \cdots - b_n \boldsymbol{w}_n.$$

Since $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$ is a basis for V, it's independent, and therefore, all of the a_i, b_j must be zero, and therefore, $\boldsymbol{v} = \boldsymbol{0}$. ■

The subspace W constructed in the theorem above is called a **complement** of U. It is not unique; indeed, it depends on the choice of basis vectors. For example, if U is a one-dimensional subspace of $\mathbb{R}^2$; that is, a line, then any other non-parallel line through the origin provides a complement of U. Later we will see that an especially useful choice of complement is the **orthogonal complement**.

Chapter 2

Linear Transformations

At an elementary level, Linear Algebra is the study of vectors (in $\mathbb{R}^n$) and matrices. Of course, much of that study revolves around systems of equations. Recall that if **x** is a vector in $\mathbb{R}^n$ (viewed as an $n \times 1$ column matrix), and A is an $m \times n$ matrix, then $\mathbf{y} = A\mathbf{x}$ is a vector in $\mathbb{R}^m$. Thus, multiplication by A produces a function from $\mathbb{R}^n$ to $\mathbb{R}^m$.

This example motivates the definition of a linear transformation, and as we'll see, provides the archetype for all linear transformations in the finite-dimensional setting. Many areas of mathematics can be viewed at some fundamental level as the study of sets with certain properties, and the functions between them. Linear algebra is no different. The sets in this context are, of course, vector spaces. Since we care about the linear algebraic structure of vector spaces, it should come as no surprise that we're most interested in functions that preserve this structure. That is precisely the idea behind linear transformations.

2.1 Definition and examples

Let V and W be vector spaces. At their most basic, all vector spaces are sets. Given any two sets, we can consider functions from one to the other. The functions of interest in linear algebra are those that respect the vector space structure of the sets.

> **Definition 2.1.1**
>
> Let V and W be vector spaces. A function $T : V \to W$ is called a **linear transformation** if:
>
> 1. For all $\mathbf{v}_1, \mathbf{v}_2 \in V$, $T(\mathbf{v}_1 + \mathbf{v}_2) = T(\mathbf{v}_1) + T(\mathbf{v}_2)$.
>
> 2. For all $\mathbf{v} \in V$ and scalars c, $T(c\mathbf{v}) = cT(\mathbf{v})$.
>
> We often use the term **linear operator** to refer to a linear transformation $T : V \to V$ from a vector space to itself.

Note on notation: it is common usage to drop the usual parentheses of function notation when working with linear transformations, as long as this does not cause confusion. That is, one might write $T\mathbf{v}$ instead of $T(\mathbf{v})$, but one should never write $T\mathbf{v} + \mathbf{w}$ in place of $T(\mathbf{v} + \mathbf{w})$, for the same reason that one should never write $2x + y$ in place of $2(x + y)$. Mathematicians often think of linear transformations in terms of matrix multiplication, which probably explains this notation to some extent.

The properties of a linear transformation tell us that a linear map T preserves the operations of addition and scalar multiplication. (When the domain and codomain

are different vector spaces, we might say that T intertwines the operations of the two vector spaces.) In particular, any linear transformation T must preserve the zero vector, and respect linear combinations.

Theorem 2.1.2

Let $T : V \to W$ be a linear transformation. Then

1. $T(\mathbf{0}_V) = \mathbf{0}_W$, and

2. For any scalars $c_1, \ldots, c_n$ and vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in V$,

$$T(c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n) = c_1 T(\mathbf{v}_1) + c_2 T(\mathbf{v}_2) + \cdots + c_n T(\mathbf{v}_n).$$

Proof.

1. *Since $\mathbf{0}_V + \mathbf{0}_V = \mathbf{0}_V$, we have*

$$T(\mathbf{0}_V) = T(\mathbf{0}_V + \mathbf{0}_V) = T(\mathbf{0}_V) + T(\mathbf{0}_V).$$

Adding $-T(\mathbf{0}_V)$ to both sides of the above gives us $\mathbf{0}_W = T(\mathbf{0}_V)$.

2. *The addition property of a linear transformation can be extended to sums of three or more vectors using associativity. Therefore, we have*

$$\begin{aligned} T(c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n) &= T(c_1\mathbf{v}_1) + \cdots T(c_n\mathbf{v}_n) \\ &= c_1 T(\mathbf{v}_1) + \cdots + c_n T(\mathbf{v}_n), \end{aligned}$$

where the second line follows from the scalar multiplication property.

∎

Example 2.1.3

Let $V = \mathbb{R}^n$ and let $W = \mathbb{R}^m$. For any $m \times n$ matrix A, the map $T_A : \mathbb{R}^n \to \mathbb{R}^m$ defined by

$$T_A(\mathbf{x}) = A\mathbf{x}$$

is a linear transformation. (This follows immediately from properties of matrix multiplication.)

Let $B = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ denote the standard basis of $\mathbb{R}^n$. Recall that $A\mathbf{e}_i$ is equal to the ith column of A. Thus, if we know the value of a linear transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ on each basis vector, we can immediately determine the matrix A such that $T = T_A$:

$$A = \begin{bmatrix} T(\mathbf{e}_1) & T(\mathbf{e}_2) & \cdots & T(\mathbf{e}_n) \end{bmatrix}.$$

This is true because T and T_A agree on the standard basis: for each $i = 1, 2, \ldots, n$,

$$T_A(\mathbf{e}_i) = A\mathbf{e}_i = T(\mathbf{e}_i).$$

Moreover, if two linear transformations agree on a basis, they must be equal. Given any $\mathbf{x} \in \mathbb{R}^n$, we can write $\mathbf{x}$ uniquely as a linear combination

$$\mathbf{x} = c_1\mathbf{e}_1 + c_2\mathbf{e}_2 + \cdots + c_n\mathbf{e}_n.$$

If $T(\mathbf{e}_i) = T_A(\mathbf{e}_i)$ for each i, then by Theorem ?? we have

$$T(\mathbf{x}) = T(c_1\mathbf{e}_1 + c_2\mathbf{e}_2 + \cdots + c_n\mathbf{e}_n)$$

$$\begin{aligned}
&= c_1 T(\boldsymbol{e}_1) + c_2 T(\boldsymbol{e}_2) + \cdots + c_n T(\boldsymbol{e}_n) \\
&= c_1 T_A(\boldsymbol{e}_1) + c_2 T_A(\boldsymbol{e}_2) + \cdots + c_n T_A(\boldsymbol{e}_n) \\
&= T_A(c_1 \boldsymbol{e}_1 + c_2 \boldsymbol{e}_2 + \cdots + c_n \boldsymbol{e}_n) \\
&= T_A(\boldsymbol{x}).
\end{aligned}$$

Let's look at some other examples of linear transformations.

- *For any vector spaces V, W we can define the **zero transformation** $0 : V \to W$ by $0(\boldsymbol{v}) = \boldsymbol{0}$ for all $\boldsymbol{v} \in V$.*

- *On any vector space V we have the **identity transformation** $1_V : V \to V$ defined by $1_V(\boldsymbol{v}) = \boldsymbol{v}$ for all $\boldsymbol{v} \in V$.*

- *Let $V = F[a, b]$ be the space of all functions $f : [a, b] \to \mathbb{R}$. For any $c \in [a, b]$ we have the **evaluation map** $E_a : V \to \mathbb{R}$ defined by $E_a(f) = f(a)$.*

  To see that this is linear, note that $E_a(0) = \boldsymbol{0}(a) = 0$, where $\boldsymbol{0}$ denotes the zero function; for any $f, g \in V$,

  $$E_a(f + g) = (f + g)(a) = f(a) + g(a) = E_a(f) + E_a(g),$$

  and for any scalar $c \in \mathbb{R}$,

  $$E_a(cf) = (cf)(a) = c(f(a)) = cE_a(f).$$

  Note that the evaluation map can similarly be defined as a linear transformation on any vector space of polynomials.

- *On the vector space $C[a, b]$ of all continuous functions on $[a, b]$, we have the integration map $I : C[a, b] \to \mathbb{R}$ defined by $I(f) = \int_a^b f(x)\, dx$. The fact that this is a linear map follows from properties of integrals proved in a calculus class.*

- *On the vector space $C^1(a, b)$ of continuously differentiable functions on (a, b), we have the differentiation map $D : C^1(a, b) \to C(a, b)$ defined by $D(f) = f'$. Again, linearity follows from properties of the derivative.*

- *Let $\mathbb{R}^\infty$ denote the set of sequences $(a_1, a_2, a_3, \ldots)$ of real numbers, with term-by-term addition and scalar multiplication. The shift operators*

  $$\begin{aligned}
  S_L(a_1, a_2, a_3, \ldots) &= (a_2, a_3, a_4, \ldots) \\
  S_R(a_1, a_2, a_3, \ldots) &= (0, a_1, a_2, \ldots)
  \end{aligned}$$

  are both linear.

- *On the space $M_{mn}(\mathbb{R})$ of $m \times n$ matrices, the trace defines a linear map $\operatorname{tr} : M_{mn}(\mathbb{R}) \to \mathbb{R}$, and the transpose defines a linear map $T : M_{mn}(\mathbb{R}) \to M_{nm}(\mathbb{R})$. The determinant and inverse operations on M_{nn} are not linear.*

For finite-dimensional vector spaces, it is often convenient to work in terms of a basis. The properties of a linear transformation tell us that we can completely define any linear transformation by giving its values on a basis. In fact, it's enough to know the value of a transformation on a spanning set. The argument given in Example ?? can be applied to any linear transformation, to obtain the following result.

> **Theorem 2.1.4**
>
> Let $T : V \to W$ and $S : V \to W$ be two linear transformations. If $V = span\{v_1, \ldots, v_n\}$ and $T(v_i) = S(v_i)$ for each $i = 1, 2, \ldots, n$, then $T = S$.

Caution: If the above spanning set is not also independent, then we can't just define the values $T(v_i)$ however we want. For example, suppose we want to define $T : \mathbb{R}^2 \to \mathbb{R}^2$, and we set $\mathbb{R}^2 = span\,(1, 2), (4, -1), (5, 1)$. If $T(1, 2) = (3, 4)$ and $T(4, -1) = (-2, 2)$, then we must have $T(5, 1) = (1, 6)$. Why? Because $(5, 1) = (1, 2) + (4, 1)$, and if T is to be linear, then we have to have $T((1, 2) + (4, -1)) = T(1, 2) + T(4, -1)$.

If for some reason we already know that our transformation is linear, we might still be concerned about the fact that if a spanning set is not independent, there will be more than one way to express a vector as linear combination of vectors in that set. If we define T by giving its values on a spanning set, will it be well-defined? (That is, could we get two different values for $T(v)$ by expressing v in terms of the spanning set in two different ways?) Suppose that we have scalars $a_1, \ldots, a_n, b_1, \ldots, b_n$ such that

$$v = a_1 v_1 + \cdots + a_n v_n \quad \text{and}$$
$$v = b_1 v_1 + \cdots + b_n v_n$$

We then have

$$a_1 T(v_1) + \cdots + a_n T(v_n) = T(a_1 v_1 + \cdots + a_n v_n)$$
$$= T(b_1 v_1 + \cdots + b_n v_n)$$
$$= b_1 T(v_1) + \cdots + b_n T(v_n).$$

Of course, we can avoid all of this unpleasantness by using a basis to define a transformation. Given a basis $B = \{v_1, \ldots, v_n\}$ for a vector space V, we can define a transformation $T : V \to W$ by setting $T(v_i) = w_i$ for some choice of vectors $w_1, \ldots, w_n$ and defining

$$T(c_1 v_1 + \cdots + c_n v_n) = c_1 T(v_1) + \cdots + c_n T(v_n).$$

Because each vector $v \in V$ can be written uniquely in terms of a basis, we know that our transformation is well-defined.

The next theorem seems like an obvious consequence of the above, and indeed, one might wonder where the assumption of a basis is needed. The distinction here is that the vectors $w_1, \ldots, w_n \in W$ are chosen in advance, and then we set $T(vecb_i) = w_i$, rather than simply defining each w_i as $T(b_i)$.

> **Theorem 2.1.5**
>
> Let V, W be vector spaces. Let $B = \{b_1, \ldots, b_n\}$ be a basis of V, and let $w_1, \ldots, w_n$ be any vectors in W. (These vectors need not be distinct.) Then there exists a unique linear transformation $T : V \to W$ such that $T(b_i) = w_i$ for each $i = 1, 2, \ldots, n$; indeed, we can define T as follows: given $v \in V$, write $v = c_1 v_1 + \cdots + c_n v_n$. Then
>
> $$T(v) = T(c_1 v_1 + \cdots + c_n v_n) = c_1 w_1 + \cdots + c_n w_n.$$

With the basic theory out of the way, let's look at a few basic examples.

Exercise 2.1.6

Suppose $T : \mathbb{R}^2 \to \mathbb{R}^2$ is a linear transformation. If $T \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}$ and

$T \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$, find $T = \begin{bmatrix} -2 \\ 4 \end{bmatrix}$.

Exercise 2.1.7

Suppose $T : \mathbb{R}^2 \to \mathbb{R}^2$ is a linear transformation. Given that $T \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$

and $T \begin{bmatrix} 2 \\ -5 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$, find $T \begin{bmatrix} 4 \\ 3 \end{bmatrix}$.

Exercise 2.1.8

Suppose $T : P_2(\mathbb{R}) \to \mathbb{R}$ is defined by

$$T(x + 2) = 1, T(1) = 5, T(x^2 + x) = 0.$$

Find $T(2 - x + 3x^2)$.

Exercise 2.1.9

Find a linear transformation $T : \mathbb{R}^2 \to \mathbb{R}^3$ such that

$$T(1, 2) = (1, 1, 0) \quad and \quad T(-1, 1) = (0, 2, -1).$$

Then, determine the value of $T(3, 2)$.

Exercise 2.1.10

Let $T : V \to W$ be a linear transformation. Prove that for any vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in V$, if $\{T(\mathbf{v}_1), \ldots, T(\mathbf{v}_n)\}$ is linearly independent in W, then $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ is linearly independent in V.

2.2 Kernel and Image

Given any linear transformation $T : V \to W$ we can associate two important subspaces: the **kernel** of T (also known as the **nullspace**), and the **image** of T (also known as the **range**).

Definition 2.2.1

Let $T : V \to W$ be a linear transformation. The **kernel** of T, denoted $\ker T$, is defined by

$$\ker T = \{\mathbf{v} \in V \mid T(\mathbf{v}) = \mathbf{0}\}.$$

The **image** of T, denoted $\operatorname{im} T$, is defined by

$$\operatorname{im} T = \{T(\mathbf{v}) \mid \mathbf{v} \in V\}.$$

Note that the kernel of T is just the set of all vectors T sends to zero. The image of T is the range of T in the usual sense of the range of a function.

> **Theorem 2.2.2**
>
> **For any linear transformation $T : V \to W$,**
>
> 1. **$\ker T$ is a subspace of V.**
>
> 2. **$\operatorname{im} T$ is a subspace of W.**

Proof.

1. To show that $\ker T$ is a subspace, first note that $\mathbf{0} \in \ker T$, since $T(\mathbf{0}) = \mathbf{0}$ for any linear transformation T. If $\mathbf{v}, \mathbf{w} \in \ker T$, then $T(\mathbf{v}) = \mathbf{0}$ and $T(\mathbf{w}) = 0$, and therefore,

$$T(\mathbf{v} + \mathbf{w}) = T(\mathbf{v}) + T(\mathbf{w}) = \mathbf{0} + \mathbf{0} = \mathbf{0}.$$

Similarly, for any scalar c and $\mathbf{v} \in \ker T$,

$$T(c\mathbf{v}) = cT(\mathbf{v}) = c\mathbf{0} = \mathbf{0}.$$

By the subspace test, $\ker T$ is a subspace.

2. Again, since $T(\mathbf{0}) = \mathbf{0}$, we see that $\mathbf{0} \in \operatorname{im} T$, so $\operatorname{im} T$ is nonempty. If $\mathbf{w}_1, \mathbf{w}_2 \in \operatorname{im} T$, then there exist $\mathbf{v}_1, \mathbf{v}_2 \in V$ such that $T(\mathbf{v}_1) = \mathbf{w}_1$ and $T(\mathbf{v}_2) = \mathbf{w}_2$. It follows that
$$\mathbf{w}_1 + \mathbf{w}_2 = T(\mathbf{v}_1) + T(\mathbf{v}_2) = T(\mathbf{v}_1 + \mathbf{v}_2),$$
so $\mathbf{w}_1 + \mathbf{w}_2 \in \operatorname{im} T$. Similarly, if c is any scalar and $\mathbf{w} = T(\mathbf{v}) \in \operatorname{im} T$, then

$$c\mathbf{w} = cT(\mathbf{v}) = T(c\mathbf{v}),$$

so $c\mathbf{w} \in \operatorname{im} T$.

∎

A familiar setting that you may already have encountered in a previous linear algebra course is that of a matrix transformation. Let A be an $m \times n$ matrix. Then we can define $T : \mathbb{R}^n \to \mathbb{R}^m$ by $T(\mathbf{x}) = A\mathbf{x}$, where elements of $\mathbb{R}^n, \mathbb{R}^m$ are considered as column vectors. We then have

$$\ker T = \operatorname{null}(A) = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{0}\}$$

and
$$\operatorname{im} T = \operatorname{col}(A) = \{A\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n\},$$

where $\operatorname{col}(A)$ denotes the **column space** of A. Recall further that if we write A in terms of its columns as
$$A = \begin{bmatrix} C_1 & C_2 & \cdots & C_n \end{bmatrix}$$

and a vector $\mathbf{x} \in \mathbb{R}^n$ as $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, then

$$A\mathbf{x} = x_1 C_1 + x_2 C_2 + \cdots + x_n C_n.$$

Thus, any element of $\operatorname{col}(A)$ is a linear combination of its columns, explaining the name column space.

Determining $null(A)$ *and* $col(A)$ *for a given matrix* A *is, unsurprisingly, a matter of reducing* A *to row-echelon form. Finding* $null(A)$ *is simply a matter of describing the set of all solutions to the homogeneous system* $A\mathbf{x} = \mathbf{0}$. *Finding* $col(A)$ *relies on the following theorem.*

Theorem 2.2.3

Let A **be an** $m \times n$ **matrix with columns** $C_1, C_2, \ldots, C_n$. **If the reduced row-echelon form of** A **has leading ones in columns** $j_1, j_2, \ldots, j_k$, **then**

$$\{C_{j_1}, C_{j_2}, \ldots, C_{j_k}\}$$

is a basis for $col(A)$.

The truth of this theorem is demonstrated in Section 5.4 of the text by Nicholson. To see why it works, we need to remember a few basic facts from elementary linear algebra. First, recall that performing an elementary row operation on a matrix A *is equivalent to multiplying on the left by an elementary matrix* E *defined using the same row operation.*

Since every elementary matrix is invertible, and any product of invertible matrices is invertible, and we can transform A *into a row-echelon matrix* R *using elementary row operations, it follows that* $R = UA$ *for an invertible matrix* U; *indeed, we have* $U = E_k E_{k-1} \cdots E_2 E_1$, *where* $E_1, \ldots, E_k$ *are the elementary matrices corresponding to the row operations used to carry* A *to* R.

A basis for $col(R)$ *is given by the columns of* R *containing the leading ones. The reason for this is as follows. First, recall that each nonzero row begins with a leading one. So if the leading ones of* R *are in columns* $i_1, \ldots, i_k$, *then there are* k *nonzero rows. Since all rows of zeros go at the bottom, each column in* R *has its last* $m - k$ *entries identically zero. Thus,*

$$col(R) \subseteq \left\{ \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^m \,\middle|\, a_1, \ldots, a_k \in \mathbb{R} \right\},$$

so $\dim col(R) \leq k$. *But the columns containing leading ones are easily shown to be independent, so they form a basis of* $col(R)$, *which therefore has dimension* $k = rank(A)$.

Next, since $R = UA$, *where* U *is invertible, if* $R = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_n \end{bmatrix}$ *and* $A = \begin{bmatrix} C_1 & C_2 & \cdots & C_n \end{bmatrix}$, *then* $C_i = U^{-1}Y_i$ *for each* i. *It follows from the fact that* U *is invertible and that the columns containing leading ones in* R *form a basis for* $col(R)$ *that the corresponding columns in* A *form a basis for* $col(A)$. *(For details, see Section 5.4 in Nicholson.)*

For example, consider the linear transformation $T : \mathbb{R}^4 \to \mathbb{R}^3$ *defined by the matrix*

$$A = \begin{bmatrix} 1 & 3 & 0 & -2 \\ -2 & -1 & 2 & 0 \\ 1 & 8 & 2 & -6 \end{bmatrix}.$$

Let's determine the RREF *of* A:

```
from sympy import Matrix,init_printing
init_printing()
A=Matrix(3,4,[1,3,0,-2,-2,-1,2,0,1,8,2,-6])
```

```
A.rref()
```

$$\begin{bmatrix} 1 & 0 & -\frac{6}{5} & \frac{2}{5} \\ 0 & 1 & \frac{2}{5} & -\frac{4}{5} \\ 0 & 0 & 0 & 0 \end{bmatrix}, (0,1)$$

We see that there are leading ones in the first and second column. Therefore,

$$\text{col}(A) = \text{im}(T) = \text{span}\left\{ \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -1 \\ 8 \end{bmatrix} \right\}. \text{ Indeed, note that}$$

$$\begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix} = -\frac{6}{5} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} + \frac{2}{5} \begin{bmatrix} 3 \\ -1 \\ 8 \end{bmatrix}$$

and

$$\begin{bmatrix} -2 \\ 0 \\ -6 \end{bmatrix} = \frac{2}{5} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} - \frac{4}{5} \begin{bmatrix} 3 \\ -1 \\ 8 \end{bmatrix},$$

so that indeed, the third and fourth columns are in the span of the first and second.

Furthermore, we can determine the nullspace: if $A\mathbf{x} = \mathbf{0}$ where $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$, then

we must have

$$x_1 = \frac{6}{5}x_3 - \frac{2}{5}x_4$$

$$x_2 = -\frac{2}{5}x_3 + \frac{4}{5}x_4,$$

so

$$\mathbf{x} = \begin{bmatrix} \frac{6}{5}x_3 - \frac{2}{5}x_4 \\ -\frac{2}{5}x_3 + \frac{4}{5}x_4 \\ x_3 \\ x_4 \end{bmatrix} = \frac{x_3}{5} \begin{bmatrix} 6 \\ -2 \\ 5 \\ 0 \end{bmatrix} + \frac{x_4}{5} \begin{bmatrix} -2 \\ 4 \\ 0 \\ 5 \end{bmatrix}.$$

It follows that a basis for $\text{null}(A) = \ker T$ is $\left\{ \begin{bmatrix} 6 \\ -2 \\ 5 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 4 \\ 0 \\ 5 \end{bmatrix} \right\}.$

Incidentally, the SymPy library for Python has built-in functions for computing nullspace and column space. But it's probably worth your while to know how to determine these from the RREF of a matrix, without additional help from the computer. That said, let's see how the computer's output compares to what we found:

```
A.nullspace()
```

$$\left[\begin{bmatrix} \frac{6}{5} \\ -\frac{2}{5} \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -\frac{2}{5} \\ \frac{4}{5} \\ 0 \\ 1 \end{bmatrix} \right]$$

```
A.columnspace()
```

$$\left[\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -1 \\ 8 \end{bmatrix} \right]$$

Note that the output from the computer simply states the basis for each space. Of course, for computational purposes, this is typically good enough.

An important result that comes out while trying to show that the "pivot columns" of a matrix (the ones that end up with leading ones in the RREF*) are a basis for the column space is that the column rank (defined as the dimension of* $\mathrm{col}(A)$*) and the row rank (the dimension of the space spanned by the columns of* A*) are equal. One can therefore speak unambiguously about the* **rank** *of a matrix* A*, and it is just as it's defined in a first course in linear algebra: the number of leading ones in the* RREF *of* A*.*

For a general linear transformation, we can't necessarily speak in terms of rows and columns, but if $T : V \to W$ *is linear, and either* V *or* W *is finite-dimensional, then we can define the rank of* T *as follows.*

> **Definition 2.2.4**
>
> *Let* $T : V \to W$ *be a linear transformation. Then the* **rank** *of* T *is defined by*
>
> $$\mathrm{rank}\, T = \dim \mathrm{im}\, T,$$
>
> *and the* **nullity** *of* T *is defined by*
>
> $$\mathrm{nullity}\, T = \dim \ker T.$$

Note that if W *is finite-dimensional, then so is* $\mathrm{im}\, T$*, since it's a subspace of* W*. On the other hand, if* V *is finite-dimensional, then we can find a basis* $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ *of* V*, and the set* $\{T(\mathbf{v}_1), \ldots, T(\mathbf{v}_n)\}$ *will span* $\mathrm{im}\, T$*, so again the image is finite-dimensional, so the rank of* T *is finite. It is possible for either the rank or the nullity of a transformation to be infinite.*

Knowing that the kernel and image of an operator are subspaces gives us an easy way to define subspaces. From the textbook, we have the following nice example.

Example 2.2.5

Let $T : M_{nn} \to M_{nn}$ *be defined by* $T(A) = A - A^T$*. Then*

1. *T is a linear map.*

2. *$\ker T$ is equal to the set of all symmetric matrices.*

3. *$\mathrm{im}\, T$ is equal to the set of all skew-symmetric matrices.*

Solution.

1. *We have $T(0) = 0$ since $0^T = 0$. Using proerties of the transpose and matrix algebra, we have*

   $$T(A+B) = (A+B) - (A+B)^T = (A - A^T) + (B - B^T) = T(A) + T(B)$$

   and

   $$T(kA) = (kA) - (kA)^T = kA - kA^T = k(A - A^T) = kT(A).$$

2. *It's clear that if $A^T = A$, then $T(A) = 0$. On the other hand, if $T(A) = 0$, then $A - A^T = 0$, so $A = A^T$. Thus, the kernel consists of all symmetric matrices.*

3. *If $B = T(A) = A - A^T$, then*

   $$B^T = (A - A^T)^T = A^T - A = -B,$$

> so certainly every matrix in im A is skew-symmetric. On the other hand,
> if B is skew-symmetric, then $B = T(\frac{1}{2}B)$, since

$$T\left(\frac{1}{2}B\right) = \frac{1}{2}T(B) = \frac{1}{2}(B - B^T) = \frac{1}{2}(B - (-B)) = B.$$

You'll recall from a course like Math 2000 that in the study of functions, the properties of being injective (one-to-one) and surjective (onto) are important. They're important for linear transformations as well, and defined in exactly the same way.

It's clear that being surjective is closely tied to image. Indeed, by definition, $T : V \to W$ is onto if $\operatorname{im} T = W$. What might not be immediately obvious is that the kernel tells us if a linear map is injective.

> **Theorem 2.2.6**
>
> Let $T : V \to W$ be a linear transformation. Then T is injective if and only if $\ker T = \{\mathbf{0}\}$.

Proof. Suppose T is injective, and let $\mathbf{v} \in \ker T$. Then $T(\mathbf{v}) = \mathbf{0}$. On the other hand, we know that $T(\mathbf{0}) = \mathbf{0}$, and since T is injective, we must have $\mathbf{v} = \mathbf{0}$. Conversely, suppose that $\ker T = \{0\}$ and that $T(\mathbf{v}_1) = T(\mathbf{v}_2)$ for some $\mathbf{v}_1, \mathbf{v}_2 \in V$. Then

$$\mathbf{0} = T(\mathbf{v}_1) - T(\mathbf{v}_2) = T(\mathbf{v}_1 - \mathbf{v}_2),$$

so $\mathbf{v}_1 - \mathbf{v}_2 \in \ker T$. Therefore, we must have $\mathbf{v}_1 - \mathbf{v}_2 = \mathbf{0}$, so $\mathbf{v}_1 = \mathbf{v}_2$, and it follows that T is injective. ∎

Let us return to the case of a matrix transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$. Notice that $\ker T_A$ is simply the set of all solutions to $A\mathbf{x} = \mathbf{0}$, while $\operatorname{im} T_A$ is the set of all $\mathbf{y} \in \mathbb{R}^m$ for which $A\mathbf{x} = \mathbf{y}$ has a solution.

Recall from the discussion above that $\operatorname{rank} A = \dim \operatorname{col}(A) = \dim \operatorname{im} T_A$. It follows that T_A is surjective if and only if $\operatorname{rank} A = m$. On the other hand, T_A is injective if and only if $\operatorname{rank} A = n$, because we know that the system $A\mathbf{x} = \mathbf{0}$ has a unique solution if and only if each column of A contains a leading one.

This has some interesting consequences. If $m = n$ (that is, if A is square), then each increase in $\dim \operatorname{null}(A)$ produces a corresponding decrease in $\dim \operatorname{col}(A)$, since both correspond to the "loss" of a leading one. Moreover, if $\operatorname{rank} A = n$, then T_A is both injective and surjective. Recall that a function is invertible if and only if it is both injective and surjective. It should come as no surprise that invertibility of T_A (as a function) is equivalent to invertibility of A (as a matrix).

Also, note that if $m < n$, then $\operatorname{rank} A \leq m < n$, so T_A could be surjective, but can't possibly be injective. On the other hand, if $m > n$, then $\operatorname{rank} A \leq n < m$, so T_A could be injective, but can't possibly be surjective. These results generalize to linear transformations between any finite-dimensional vector spaces. The first step towards this is the following theorem, which is sometimes known as the Fundamental Theorem of Linear Transformations.

> **Theorem 2.2.7 Dimension Theorem.**
>
> Let $T : V \to W$ be any linear transformation such that $\ker T$ and $\operatorname{im} T$ are finite-dimensional. Then V is finite-dimensional, and
>
> $$\dim V = \dim \ker T + \dim \operatorname{im} T.$$

Proof. *The trick with this proof is that we aren't assuming V is finite-dimensional, so we can't start with a basis of V. But we do know that $\operatorname{im} T$ is finite-dimensional, so we start with a basis $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\}$ of $\operatorname{im} T$. Of course, every vector in $\operatorname{im} T$ is the image of some vector in V, so we can write $\boldsymbol{w}_i = T(\boldsymbol{v}_i)$, where $\boldsymbol{v}_i \in V$, for $i = 1, 2, \ldots, m$.*

Since $\{T(\boldsymbol{v}_1), \ldots, T(\boldsymbol{v}_m)\}$ is a basis, it is linearly independent. The results of Exercise ?? tell us that the set $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ must therefore be independent.

We now introduce a basis $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ of $\ker T$, which we also know to be finite-dimensional. If we can show that the set $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is a basis for V, we'd be done, since the number of vectors in this basis is $\dim \ker T + \dim \operatorname{im} T$. We must therefore show that this set is independent, and that it spans V.

To see that it's independent, suppose that

$$a_1\boldsymbol{u}_1 + \cdots + a_n\boldsymbol{u}_n + b_1\boldsymbol{v}_1 + \cdots + b_m\boldsymbol{v}_m = \boldsymbol{0}.$$

Applying T to this equation, and noting that $T(\boldsymbol{u}_i) = \boldsymbol{0}$ for each i, by definition of the $\boldsymbol{u}_i$, we get

$$b_1 T(\boldsymbol{v}_1) + \cdots + b_m T(\boldsymbol{v}_m) = \boldsymbol{0}.$$

We assumed that the vectors $T(\boldsymbol{v}_i)$ were independent, so all the b_i must be zero. But then we get

$$a_1\boldsymbol{u}_1 + \cdots + a_n\boldsymbol{u}_n = \boldsymbol{0},$$

and since the $\boldsymbol{u}_i$ are independent, all the a_i must be zero.

To see that these vectors span, choose any $\boldsymbol{x} \in V$. Since $T(\boldsymbol{x}) \in \operatorname{im} T$, there exist scalars $c_1, \ldots, c_m$ such that

$$T(\boldsymbol{x}) = c_1 T(\boldsymbol{v}_1) + \cdots + c_m T(\boldsymbol{v}_m). \tag{2.2.1}$$

We'd like to be able to conclude from this that $\boldsymbol{x} = c_1\boldsymbol{v}_1 + \cdots + c_m\boldsymbol{v}_m$, but this would be false, unless T was known to be injective (which it isn't). Failure to be injective involves the kernel -- how do we bring that into the picture?

The trick is to realize that the reason we might have $\boldsymbol{x} \neq c_1\boldsymbol{v}_1 + \cdots + c_m\boldsymbol{v}_m$ is that we're off by something in the kernel. Indeed, (??) can be re-written as

$$T(\boldsymbol{x} - c_1\boldsymbol{v}_1 - \cdots - c_m\boldsymbol{v}_m) = \boldsymbol{0},$$

so $\boldsymbol{x} - c_1\boldsymbol{v}_1 - \cdots - c_m\boldsymbol{v}_m \in \ker T$. But we have a basis for $\ker T$, so we can write

$$\boldsymbol{x} - c_1\boldsymbol{v}_1 - \cdots - c_m\boldsymbol{v}_m = t_1\boldsymbol{u}_1 + \cdots + t_n\boldsymbol{u}_n$$

for some scalars $t_1, \ldots, t_n$, and this can be rearranged to give

$$\boldsymbol{x} = t_1\boldsymbol{u}_1 + \cdots + t_n\boldsymbol{u}_n + c_1\boldsymbol{v}_1 + \cdots + c_m\boldsymbol{v}_m,$$

which completes the proof. ∎

This is sometimes known as the Rank-Nullity Theorem, since it can be stated in the form

$$\dim V = \operatorname{rank} T + \operatorname{nullity} T.$$

We will see that this result is frequently useful for providing simple arguments that establish otherwise difficult results. A basic situation where the theorem is useful is as follows: we are given $T : V \to W$, where the dimensions of V and W are known. Since $\operatorname{im} T$ is a subspace of W, we know from Theorem ?? that T is onto if and only if $\dim \operatorname{im} T = \dim W$. In many cases it is easier to compute $\ker T$ than it is $\operatorname{im} T$, and the Dimension Theorem lets us determine $\dim \operatorname{im} T$ if we know $\dim V$ and $\dim \ker T$.

A useful consequence of this result is that if we know V is finite-dimensional, we can order any basis such that the first vectors in the list are a basis of $\ker T$, and the

images of the remaining vectors produce a basis of $\operatorname{im} T$.

Note that one consequence of the dimension theorem is that we must have

$$\dim \ker T \leq \dim V \quad \text{and} \quad \dim \operatorname{im} T \leq \dim V.$$

Of course, we must also have $\dim \operatorname{im} T \leq \dim W$, since $\operatorname{im} T$ is a subspace of W. In the case of a matrix transformation T_A, this means that the rank of T_A is at most the minimum of $\dim V$ and $\dim W$. This once again has consequences for the existence and uniqueness of solutions for linear systems with the coefficient matrix A.

Exercise 2.2.8

Let V and W be finite-dimensional vector spaces. Prove the following:

1. $\dim V \leq \dim W$ if and only if there exists an injection $T : V \to W$.

2. $\dim V \geq \dim W$ if and only if there exists a surjection $T : V \to W$.

2.3 Isomorphisms (a.k.a. invertible linear maps)

We ended the last section with an important result. Exercise **??** showed that existence of an injective linear map $T : V \to W$ is equivalent to $\dim V \leq \dim W$, and that existence of a surjective linear map is equivalent to $\dim V \geq \dim W$. It's probably not surprising than that existence of a bijective linear map $T : V \to W$ is equivalent to $\dim V = \dim W$.

Definition 2.3.1

A bijective linear transformation $T : V \to W$ is called an **isomorphism**. If such a map exists, we say that V and W are **isomorphic**, and write $V \cong W$.

Theorem 2.3.2

For any finite-dimensional vector spaces V and W, $V \cong W$ if any only if $\dim V = \dim W$**.**

Proof. If $T : V \to W$ is a bijection, then it is both injective and surjective. Since T is injective, $\dim V \leq \dim W$, by Exercise **??**. By this same exercise, since T is surjective, we must have $\dim V \geq \dim W$. It follows that $\dim V = \dim W$.

Suppose now that $\dim V = \dim W$. Then we can choose bases $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of V, and $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ of W. Theorem **??** then guarantees the existence of a linear map $T : V \to W$ such that $T(\mathbf{v}_i) = \mathbf{w}_i$ for each $i = 1, 2, \ldots, n$. Repeating the arguments of Exercise **??** shows that T is a bijection. ∎

Note that buried in the theorem above is the following useful fact: an isomorphism $T : V \to W$ takes any basis of V to a basis of W. Another remarkable result of the above theorem is that any two vector spaces of the same dimension are isomorphic! In particular, we have the following theorem.

Theorem 2.3.3

If $\dim V = n$, then $V \cong \mathbb{R}^n$.

This theorem is a direct consequence of Theorem **??**. But it's useful to understand how it works in practice.

> **Definition 2.3.4**
>
> Let V be a finite-dimensional vector space, and let $B = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ be a basis for V. The **coefficient isomorphism** associated to B is the map $C_B : V \to \mathbb{R}^n$ defined by
>
> $$C_B(c_1\boldsymbol{e}_1 + c_2\boldsymbol{e}_2 + \cdots + c_n\boldsymbol{e}_n) = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

Note that this is a well-defined map since every vector in V can be written uniquely in terms of the basis B.

The coefficient isomorphism is especially useful when we want to analyze a linear map computationally. Suppose we're given $T : V \to W$ where V, W are finite-dimensional. Let us choose bases $B = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ of V and $B' = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\}$ of W. The choice of these two bases determines scalars $a_{ij}, 1 \leq i \leq n, 1 \leq j \leq m$, such that

$$T(\boldsymbol{v}_j) = a_{1j}\boldsymbol{w}_1 + a_{2j}\boldsymbol{w}_2 + \cdots + a_{mj}\boldsymbol{w}_j,$$

for each $i = 1, 2, \ldots, n$. The resulting matrix $A = [a_{ij}]$ defines a matrix transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$T_A \circ C_B = C_{B'} \circ T.$$

The relationship among the four maps used here is best captured by the "commutative diagram" in Figure **??**.

With this connection between linear maps (in general) and matrices, it can be worthwhile to pause and consider invertibility in the context of matrices. Recall that an $n \times n$ matrix A is invertible if there exists a matrix A^{-1} such that $AA^{-1} = I_n$ and $A^{-1}A = I_n$.

The same definition can be made for linear maps. We've defined what it means for a map $T : V \to W$ to be invertible as a function. In particular, we relied on the fact that any bijection has an inverse.

First, a note on notation. Given linear maps $U \xrightarrow{T} V \xrightarrow{S} W$, we typically write the composition $S \circ T : U \to W$ as a "product" ST. The reason for this is again to mimic the case of matrices. Let A be an $m \times n$ matrix, and let B be an $n \times k$ matrix. Then we have linear maps

$$\mathbb{R}^k \xrightarrow{T_B} \mathbb{R}^n \xrightarrow{T_A} \mathbb{R}^m,$$

and the composition $T_A \circ T_B : \mathbb{R}^k \to \mathbb{R}^m$ satisfies

$$T_A \circ T_B(\boldsymbol{x}) = T_A(T_B(\boldsymbol{x})) = T_A(B\boldsymbol{x}) = A(B\boldsymbol{x}) = (AB)\boldsymbol{x} = T_{AB}(\boldsymbol{x}).$$

Note that the rules given in elementary linear algebra, for the relative sizes of matrices that can be multiplied, are simply a manifestation of the fact that to compose functions, the range of the first must be contained in the domain of the second.

> **Exercise 2.3.6**
>
> Show that the composition of two linear maps is again a linear map.

Moreover, Theorem **??** tells us why we can only consider invertibility for square matrices: we know that invertible linear maps are only defined between spaces of equal dimension. In analogy with matrices, some texts will define a linear map $T : V \to W$ to be invertible if there exists a linear map $S : W \to V$ such that

$$ST = 1_V \quad \text{and} \quad TS = 1_W.$$

$$
\begin{array}{ccc}
V & \xrightarrow{\ \ T\ \ } & W \\
{\scriptstyle C_B}\downarrow & & \downarrow{\scriptstyle C_{B'}} \\
\mathbb{R}^n & \xrightarrow{\ \ T_A\ \ } & \mathbb{R}^m
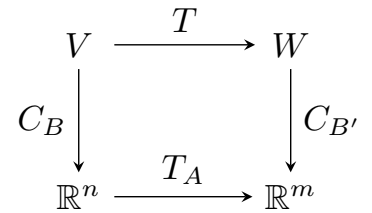\end{array}
$$

Figure 2.3.5 *Defining the matrix of a linear map with respect to choices of basis.*

From this definition, one can show that S and T must be bijections, and of course, if T is a bijection (as in our definition) then we know it has an inverse. What remains to be seen is that this inverse is also a linear map.

Exercise 2.3.7

Let $T : V \to W$ be a bijective linear transformation. Show that $T^{-1} : W \to V$ is a linear transformation.

Exercise 2.3.8

Show that if $ST = 1_V$, then S is surjective and T is injective. Conclude that if $ST = 1_V$ and $TS = 1_w$, then S and T are both bijections.

Hint. *This is really a Math 2000 problem.*

2.4 Worksheet: matrix transformations

This worksheet deals with matrix transformations, and in particular, kernel and image. The goal is to understand these important subspaces in a familiar context.

Let A be an $m \times n$ matrix. We can use A to define a transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$ given by $T_A(\mathbf{x}) = A\mathbf{x}$, where we view $\mathbf{x}$ as an $n \times 1$ column vector.

The **kernel** of T_A is the set of vectors $\mathbf{x}$ such that $T_A(\mathbf{x}) = \mathbf{0}$. That is, $\ker T_A$ is the set of solutions to the homogeneous system $A\mathbf{x} = \mathbf{0}$.

The **image** of T_A (also known as the range of T_A) is the set of vectors $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{y} = A\mathbf{x}$ for some $\mathbf{x} \in \mathbb{R}^n$. In other words, $\text{im}(T_A)$ is the set of vectors $\mathbf{y}$ for which the non-homogeneous system $A\mathbf{x} = \mathbf{y}$ is consistent.

Because T_A is a linear transformation, we can compute it as long as we're given its values on a basis. If $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ is a basis for $\mathbb{R}^n$, then for any $\mathbf{x} \in \mathbb{R}^n$ there exist unique scalars $c_1, c_2, \ldots, c_n$ such that

$$\mathbf{x} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n,$$

and since T_A is linear, we have

$$T_A(\mathbf{x}) = c_1 T_A(\mathbf{v}_1) + c_2 T_A(\mathbf{v}_2) + \cdots + c_n T_A(\mathbf{v}_n).$$

The main challenge, computationally speaking, is that if our basis is not the standard basis, some effort will be required to write $\mathbf{x}$ in terms of the given basis.

1. Confirm that

$$B = \left\{ \begin{bmatrix} 1 \\ 0 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 0 \\ -3 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \\ -3 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \\ -2 \\ 1 \end{bmatrix} \right\}$$

is a basis for $\mathbb{R}^4$.

To assist with solving this problem, a code cell is provided below. Recall that you can enter the matrix $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ as `Matrix([[a,b,c],[d,e,f`

or as `Matrix(3,3,[a,b,c,d,e,f,g,h,i])`.

The reduced row-echeleon form of `A` is given by `A.rref()`. The product of matrices `A` and `B` is simply `A*B`. The inverse of a matrix `A` can be found using `A.inv()` or simply `A**(-1)`.

One note of caution: in the HTML worksheet, if you don't import `sympy` as your first line of code, you'll instead use Sage syntax. Sage uses `A.inverse()` instead of `A.inv()`.

In a Jupyter notebook, remember you can generate additional code cells by clicking on the + button.

```
from sympy import Matrix,init_printing
init_printing()
```

You can also use the cell below to write down any necessary explanation.

2. Write each of the standard basis vectors in terms of this basis.
      Suggestion: in each case, this can be done by solving a matrix equation, using the inverse of an appropriate matrix.

A linear transformation $T : \mathbb{R}^4 \to \mathbb{R}^4$ is now defined as follows:

$$T\left(\begin{bmatrix} 1 \\ 0 \\ 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} 3 \\ 0 \\ 2 \\ -1 \end{bmatrix}, T\left(\begin{bmatrix} 4 \\ 2 \\ 0 \\ -3 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 5 \end{bmatrix}, T\left(\begin{bmatrix} 0 \\ 4 \\ -3 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} 4 \\ 2 \\ 2 \\ 4 \end{bmatrix}, T\left(\begin{bmatrix} 3 \\ 5 \\ -2 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 2 \\ 4 \\ 0 \\ 10 \end{bmatrix}.$$

Let $\{e_1, e_2, e_3, e_4\}$ denote the standard basis for $\mathbb{R}^4$.

3.   *Determine $T(e_i)$ for $i = 1, 2, 3, 4$, and in so doing, determine the matrix A such that $T = T_A$.*

4.   *Let M be the matrix whose columns are given by the values of T on the basis B. (This would be the matrix of T if B was actually the standard basis.) Let N be the matrix whose inverse you used to solve part (b). Can you find a way to combine these matrices to obtain the matrix A? If so, explain why your result makes sense.*

Next we will compute the kernel and image of the transformation from the previous exercises. Recall that when solving a homogeneous system $A\mathbf{x} = \mathbf{0}$, we find the RREF of A, and any variables whose columns do not contain a leading 1 are assigned as parameters. We then express the general solution x in terms of those parameters.

The image of a matrix transformation T_A is also known as the **column space** of A, because the range of T_A is precisely the span of the columns of A. The RREF of A tells us which columns to keep: the columns of A that correspond to the columns in the RREF of A with a leading 1.

Let T be the linear transformation given in the previous exercises.

5. Determine the kernel of T.

6. Determine the image of T.

7. The **Dimension Theorem** states that for a linear transformation $T : V \to W$, where V is finite-dimensional,

$$dim\, V = dim\, ker(T) + dim\, im(T).$$

Explain why this result makes sense using your results for this problem.

2.5 Worksheet: linear recurrences

For this worksheet, the reader is directed to Section 7.5 of Linear Algebra with Applications, by Keith Nicholson. You will probably want to read this section before proceeding.

A linear recurrence of length k is a sequence $[x_n]$ that is recursively defined, with successive terms in the sequence defined in terms of the previous k terms, via a linear recursion formula of the form

$$x_{n+k} = a_0 x_k + a_1 x_{k+1} + \cdots + a_{k-1} x_{n+k-1}.$$

(Here we assume $a_0 \neq 0$ to have the appropriate length.) The most famous example of a linear recurrence is, of course, the Fibonacci sequence, which is defined by $x_0 = 1$, $x_1 = 1$, and $x_{n+2} = x_n + x_{n+1}$ for all $n \geq 0$.

As demonstrated in the Nicholson's book, the set of all sequences satisfying a linear recursion of length k form a subspace V of the vector space $\mathbb{R}^\infty$ of all real-valued sequences. Since each sequence is determined by the k initial conditions $x_0, x_1, \ldots, x_{k-1}$, we see that V is isomorphic to $\mathbb{R}^k$.

The goal of this worksheeet is to understand how to obtain closed form expressions for a recursively defined sequence using linear algebra. That is, rather than having to generate terms of the sequence one-by-one using the recursion formula, we want a function of n that will produce each term x_n in the sequence.

Since we know the dimension of the space V of solutions, it suffices to understand two things:

- *How to produce a basis for V.*

- *How to write a given solution in terms of that basis.*

The key observation is that for each recursion relation, there is an associated polynomial. For the recursion

$$x_{n+k} = a_0 x_n + a_1 x_{n+1} + \cdots + a_{k-1} x_{n+k-1},$$

the associated polynomial is

$$p(x) = x^k - a_{k-1} x^{k-1} - \cdots - a_1 x - a_0.$$

The main result is Theorem 7.5.4 in Nicholson. If we can factor $p(x)$ completely over the reals as

$$p(x) = (x - \lambda_1)^{m_1} (x - \lambda_2)^{m_2} \cdots (x - \lambda_p)^{m_p},$$

then a basis for the space of solutions is given by

$$[\lambda_1^n], [n\lambda_1^n], \ldots, [n^{m_1-1}\lambda_1^n]$$
$$[\lambda_2^n], [n\lambda_2^n], \ldots, [n^{m_2-1}\lambda_2^n]$$
$$\vdots$$
$$[\lambda_p^n], [n\lambda_p^n], \ldots, [n^{m_p-1}\lambda_p^n].$$

Once we have a basis, we can apply the given coefficients to determine how to write a particular sequence as a linear combination of the basis vectors.

1.   Find a basis for the space V of sequences $[x_n)$ satisfying the recurrence

$$x_{n+3} = -2x_n + x_{n+1} + 2x_{n+2}.$$

Then find a formula for the sequence satisfying the initial conditions $x_0 = 1, x_1 = 2, x_2 = 1$.

To solve this problem, you may use Python code, as outlined below. To get started, load the functions you'll need from the SymPy library.

```
from sympy import symbols,factor,init_printing
x = symbols('x')
init_printing()
```

First, determine the associated polynomial for the recurrence.

(Input your polynomial in the cell below. To get proper formatting, wrap your math in $ delimiters, and can use ^ to enter exponents.)Next, factor the polynomial. You can do this using the `factor()` command. In Python, you will need to enter `**` for the exponents.In the cell below, list the roots of the polynomial, and the resulting basis B for the space V of solutions. Recall that if λ is a root of the polynomial, then $[\lambda^n)$ will be a basis vector for the vector space V of solutions. You may wish to confirm that each of your basis sequences indeed satisfies our recursion.

Next, let $s = [x_n)$ be the recursion that satisfies the given initial conditions. We want to write $[x_n)$ in terms of the basis we just found. Recall, from Nicholson, that since our basis has three elements, there is an isomorphism $T : \mathbb{R}^3 \to V$, where $T(a, b, c)$ is equal to the sequence $[x_n)$ in V that satisfies the initial conditions $x_0 = a, x_1 = b, x_2 = c$. Thus, our desired sequence is given by $s = T(1, 2, 1)$.

Let $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^3$ be the vectors such that $B = \{T(\mathbf{v}_1), T(\mathbf{v}_2), T(\mathbf{v}_3)\}$. (That is, write out the first three terms in each sequence in your basis to get three vectors.) We then need to find scalars c_1, c_2, c_3 such that

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + c_3 \mathbf{v}_3 = (1, 2, 1).$$

We will then have

$$s = T(1, 2, 1)$$
$$= c_1 T(\mathbf{v}_1) + c_2 T(\mathbf{v}_2) + c_3 T(\mathbf{v}_3),$$

and we recall that the sequences $T(\mathbf{v}_i)$ are the sequences in our basis B.

Set up this system, and then use the computer to solve. Let A be the coefficient matrix for the system, which you will need to input into the cell below, and let B be the column vector containing the initial conditions.

```
from sympy import Matrix
A = Matrix()
B = Matrix([1,2,1])
X = A**(-1)*B
X
```

Using the solution above, state the answer to this exercise. Now, we leave you with a few more exercises. Recall that if the associated polynomial for your recursion has a repeated root $(x - \lambda)^k$, then your basis will include the sequences $[\lambda^n), [n\lambda^n), \ldots, [n^{k-1}\lambda^n)$.

2.   Find a basis for the space V of sequences $[x_n)$ satisfying the recurrence

$$x_{n+3} = -4x_n + 3x_{n+2}.$$

Then find a formula for the sequence satisfying the initial conditions $x_0 = 1, x_1 = -1, x_2 = 1$.

3.   Find a basis for the space V of sequences $[x_n)$ satisfying the recurrence

$$x_{n+3} = 8x_n - 12x_{n+1} + 6x_{n+2}.$$

Then find a formula for the sequence satisfying the initial conditions $x_0 = 1, x_1 = -1, x_2 = 1$.

Our last exercise is more theoretical, and less computational. Recall that the shift operator S on the vector space $\mathbb{R}^\infty$ of sequences is the operator that deletes the first entry of a sequence. (So the remaining entries get "shifted" once to the left.)

4.   Show that the shift operator is onto, but not one-to-one. Then, determine the kernel of the shift operator.

Chapter 3

Orthogonality and Applications

3.1 Orthogonal sets of vectors

You may recall from elementary linear algebra, or a calculus class, that vectors in $\mathbb{R}^2$ or $\mathbb{R}^3$ are considered to be quantities with both magnitude and direction. Interestingly enough, neither of these properties is inherent to a general vector space. The vector space axioms specify only algebra; they say nothing about geometry. (What, for example, should be the "angle" between two polynomials?)

Because vector algebra is often introduced as a consequence of geometry (like the "tip-to-tail" rule), you may not have thought all that carefully about what, exactly, is responsible for making the connection between algebra and geometry. It turns out that the missing link is the humble dot product. This should be plausible after a bit of thought. After all, you probably encountered the following result, perhaps as a consequence of the law of cosines: for any two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^2$,

$$\boldsymbol{u} \cdot \boldsymbol{v} = \|\boldsymbol{u}\| \, \|\boldsymbol{v}\| \cos \theta,$$

where θ is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$. Here we see both magnitude and direction (encoded by the angle) defined in terms of the dot product.

While it is possible to generalize the idea of the dot product to something called an inner product, we will first focus on the basic dot product in $\mathbb{R}^n$. Once we have a good understanding of things in that setting, we can move on to consider the abstract counterpart.

3.1.1 Basic definitions and properties

For most of this chapter (primarily for typographical reasons) we will denote elements of $\mathbb{R}^n$ as ordered n-tuples $(x_1, \ldots, x_n)$ rather than as column vectors.

Definition 3.1.1

*Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ be vectors in $\mathbb{R}^n$. The **dot product** of $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted by $\boldsymbol{x} \cdot \boldsymbol{y}$ is the scalar defined by*

$$\boldsymbol{x} \cdot \boldsymbol{y} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n.$$

*The **norm** of a vector $\boldsymbol{x}$ is denoted $\|\boldsymbol{x}\|$ and defined by*

$$\|\boldsymbol{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Note that both the dot product and the norm produce scalars. Through the Pythagorean Theorem, we recognize the norm as the length of $\boldsymbol{x}$. The dot product can still be thought of as measuring the angle between vectors, although the simple geometric proof used in two dimensions is not that easily translated to n dimensions. At the very least, the dot product lets us extend the notion of right angles to higher dimensions.

> **Definition 3.1.2**
>
> We say that two vectors $x, y \in \mathbb{R}^n$ are **orthogonal** if $x \cdot y = 0$.

It should be no surprise that all the familiar properties of the dot product work just as well in any dimension. The folowing properties are all easily confirmed by routine computation.

> **Theorem 3.1.3**
>
> **For any vectors $x, y, z \in \mathbb{R}^n$,**
>
> 1. $x \cdot y = y \cdot x$
>
> 2. $x \cdot (y + z) = x \cdot y + x \cdot z$
>
> 3. **For any scalar** c, $x \cdot (cy) = (cx) \cdot y = c(x \cdot y)$
>
> 4. $x \cdot x \geq 0$, **and $x \cdot x = 0$ if and only if $x = 0$**

The above properties, when properly abstracted, become the defining properties of a (real) inner product. (A complex inner product also involves complex conjugates.) For a general inner product, the requirement $x \cdot x \geq 0$ is referred to as being positive-definite, and the property that only the zero vector produces zero when dotted with itself is called nondegenerate. Note that we have the following connection between norm and dot product:

$$\|x\|^2 = x \cdot x.$$

For a general inner product, this can be used as a definition of the norm associated to an inner product.

> **Exercise 3.1.4**
>
> Given that $\|x\| = 3$, $\|y\| = 1$, and $x \cdot y = -2$, compute $(4x - 3y) \cdot (x + 5y)$.

> **Exercise 3.1.5**
>
> Show that for any vectors $x, y \in \mathbb{R}^n$, we have
>
> $$\|x + y\|^2 = \|x\|^2 + 2x \cdot y + \|y\|^2.$$

> **Exercise 3.1.6**
>
> Suppose $\mathbb{R}^n = \text{span}\{v_1, v_2, \ldots, v_k\}$. Prove that $x = 0$ if and only if $x \cdot v_i = 0$ for each $i = 1, 2, \ldots, k$.

There are two important inequalities associated to the dot product and norm. We state them both in the following theorem.

> **Theorem 3.1.7**
>
> **Let x, y be any vectors in $\mathbb{R}^n$. Then**
>
> 1. $|x \cdot y| \leq \|x\| \|y\|$
>
> 2. $\|x + y\| \leq \|x\| + \|y\|$

The first of the above inequalities is called the Cauchy-Schwarz inequality, which

be viewed as a manifestation of the formula

$$\mathbf{x}\cdot\mathbf{y} = \|\mathbf{x}\|\|\mathbf{y}\|\cos\theta,$$

since after all, $|\cos\theta| \leq 1$ for any angle θ. For a direct proof, we note that for any scalars r, s we have

$$\|r\mathbf{x} \pm s\mathbf{y}\|^2 = r^2\|\mathbf{x}\|^2 \pm 2rs\mathbf{x}\cdot\mathbf{y} + s^2\|\mathbf{y}\|^2.$$

Putting $r = \|\mathbf{y}\|, s = \|\mathbf{x}\|$ gives

$$\|s\mathbf{x} \pm r\mathbf{y}\|^2 = 2rs(rs \pm \mathbf{x}\cdot\mathbf{y}).$$

Since the left-hand side is non-negative, we must have

$$\mathbf{x}\mathbf{y} \leq \|\mathbf{x}\|\|\mathbf{y}\| \quad \text{and} \quad -\|\mathbf{x}\|\|\mathbf{y}\| \leq \mathbf{x}\cdot\mathbf{y},$$

and the result follows.

The second result, called the triangle inequality, follows immediately from the Cauchy-Scwarz inequality and Exercise **??**. The triangle inequality gets its name from the "tip-to-tail" picture for vector addition. Essentially, it tells us that the length of any side of a triangle must be less than the sum of the lengths of the other two sides. The importance of the triangle inequality is that it tells us that the norm can be used to define distance.

> **Definition 3.1.8**
>
> For any vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the **distance** from $\mathbf{x}$ to $\mathbf{y}$ is denoted $d(\mathbf{x}, \mathbf{y})$, and defined as
> $$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|.$$

Using properties of the norm, we can show that this distance function meets the criteria of what's called a metric. A metric is any function that takes a pair of vectors (or points) as input, and returns a number as output, with the following properties:

1. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for any $\mathbf{x}, \mathbf{y}$

2. $d(\mathbf{x}, \mathbf{y}) \geq 0$, and $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$

3. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ for any $\mathbf{x}, \mathbf{y}, \mathbf{z}$

We leave it as an exercise to confirm that the distance function defined above is a metric.

In more advanced courses (e.g. topology or analysis) you might go into detailed study of these structures. There are three interrelated structures: inner products, norms, and metrics. You might consider questions like: does every norm come from an inner product? Does every metric come from a norm? (No.) Things get even more interesting for infinite-dimensional spaces. Of special interest are spaces such as Hilbert spaces (a special type of infinite-dimensional inner product space) and Banach spaces (a special type of infinite-dimensional normed space).

3.1.2 Orthogonal sets of vectors

In earlier chapters, we've seen that among different sets of vectors one could consider, independent sets and spanning sets are both worthy of study. One of the main themes of this chapter is that orthogonal sets are equally worthy, and in many cases, easier to work with.

Definition 3.1.9

A set of vectors $\{v_1, v_2, \ldots, v_k\}$ in $\mathbb{R}^n$ is called **orthogonal** if:

- $v_i \neq 0$ for each $i = 1, 2 \ldots, k$

- $v_i \cdot v_j = 0$ for all $i \neq j$

Exercise 3.1.10

Show that the following is an orthogonal subset of $\mathbb{R}^4$.

$$\{(1, 0, 1, 0), (-1, 0, 1, 1), (1, 1, -1, 2)\}$$

Can you find a fourth vector that is orthogonal to each vector in this set?

The requirement that the vectors in an orthogonal set be nonzero is partly because the alternative would be boring, and partly because it lets us state the following theorem.

Theorem 3.1.11

Any orthogonal set of vectors is linearly independent.

Proof. *Suppose* $S = \{v_1, v_2, \ldots, v_k\}$ *is orthogonal, and suppose*

$$c_1 v_1 + c_2 v_2 + \cdots + c_k v_k = 0$$

for scalars $c_1, c_2, \ldots, c_k$. *Taking the dot product of both sides of the above equation with* v_1 *gives*

$$c_1(v_1 \cdot v_1) + c_2(v_1 \cdot v_2) + \cdots + c_k(v_1 \cdot v_k) = v_1 \cdot 0$$
$$c_1 \|v_1\|^2 + 0 + \cdots + 0 = 0.$$

Since $\|v_1\|^2 \neq 0$, *we must have* $c_1 = 0$. *We similarly find that all the remaining scalars are zero by taking the dot product with* $v_2, \ldots, v_k$. ∎

Another useful consequence of orthogonality: recall that in two dimensions, we have the Pythagorean Theorem for right-angled triangles, but have to settle for the Law of Cosines otherwise. In n dimensions, we have the following, which follows from the fact that all "cross terms" (dot products of different vectors) will vanish.

Theorem 3.1.12

For any orthogonal set of vectors $\{x_1, \ldots, x_k\}$ **we have**

$$\|x_1 + \cdots + x_k\|^2 = \|x_1\|^2 + \cdots + \|x_k\|^2.$$

Our final initial result about orthogonal sets of vectors relates to span. In general, we know that if $y \in \text{span}\{x_1, \ldots, x_k\}$, then it is possible to solve for scalars $c_1, \ldots, c_k$ such that $y = c_1 x_1 + \cdots + c_k x_k$. The trouble is that finding these scalars generally involves setting up, and then solving, a system of linear equations. The great thing about orthogonal sets of vectors is that we can provide explicit formulas for the scalars.

> ### Theorem 3.1.13 Fourier expansion theorem.
>
> Let $S = \{v_1, v_2, \ldots, v_k\}$ be an orthogonal set of vectors. For any $y \in$ span S, we have
>
> $$y = \left(\frac{y \cdot v_1}{v_1 \cdot v_1}\right) v_1 + \left(\frac{y \cdot v_2}{v_2 \cdot v_2}\right) v_2 + \cdots + \left(\frac{y \cdot v_k}{v_k \cdot v_k}\right) v_k.$$

Proof. *Let $y = c_1 v_1 + \cdots + c_k v_k$. Taking the dot product of both sides of this equation with v_i gives*

$$v_i \cdot y = c_i (v_i \cdot v_i),$$

since the dot product of v_i with v_j for $i \neq j$ is zero. ∎

One use of Theorem **??** is determining whether or not a given vector is in the span of an orthogonal set. If it is in the span, then its coefficients must satisfy the Fourier expansion formula. Therefore, if we compute the right hand side of the above formula and do not get our original vector, then that vector must not be in the span.

> ### Exercise 3.1.14
>
> Determine whether or not the vectors $v = (1, -4, 3, -11), w = (3, 1, -4, 2)$ belong to the span of the vectors $x_1 = (1, 0, 1, 0), x_2 = (-1, 0, 1, 1), x_3 = (1, 1, -1, 2)$. (We confirmed that these vectors form an orthogonal set in Exercise **??**.)

The Fourier expansion is especially simple if our basis vectors have norm one, since the denominators in each coefficient disappear. Recall (from elementary linear algebra) that for any nonzero vector v, a unit vector (that is, a vector of norm one) in the direction of v is given by

$$\hat{u} = \frac{1}{\|v\|} v.$$

We often say that the vector u is normalized. (The convention of using a "hat" for unit vectors is common but not universal.)

> ### Definition 3.1.15
>
> A basis B of $\mathbb{R}^n$ is called an **orthonormal basis** if B is orthogonal, and all the vectors in B are unit vectors.

> ### Example 3.1.16
>
> In Exercise **??** we saw that the set
>
> $$\{(1, 0, 1, 0), (-1, 0, 1, 1), (1, 1, -1, 2), (1, -6, -1, 2)\}$$
>
> is orthogonal. Since it's orthogonal, it must be independent, and since it's a set of four independent vectors in $\mathbb{R}^4$, it must be a basis. To get an orthonormal basis, we normalize each vector:
>
> $$\hat{u}_1 = \frac{1}{\sqrt{1^2 + 0^2 + 1^2 + 0^2}} (1, 0, 1, 0) = \frac{1}{\sqrt{2}}(1, 0, 1, 0)$$
>
> $$\hat{u}_2 = \frac{1}{\sqrt{(-1)^2 + 0^2 + 1^2 + 1^2}} (-1, 0, 1, 1,) = \frac{1}{\sqrt{3}}(-1, 0, 1, 1)$$
>
> $$\hat{u}_3 = \frac{1}{\sqrt{1^2 + 1^2 + (-1)^2 + 2^2}} (1, 1, -1, 2) = \frac{1}{\sqrt{7}}(1, 1, -1, 2)$$

$$\hat{u}_4 = \frac{1}{\sqrt{1^2 + (-6)^2 + (-1)^2 + 2^2}}(1, -6, -1, 2) = \frac{1}{\sqrt{42}}(1, -6, -1, 2).$$

The set $\{\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4\}$ is then an orthonormal basis of $\mathbb{R}^4$.

The process of creating unit vectors does typically introduce square root coefficients in our vectors. This can seem undesirable, but there remains value in having an orthonormal basis. For example, suppose we wanted to write the vector $v = (3, 5, -1, 2)$ in terms of our basis. We can quickly compute

$$v \cdot \hat{u}_1 = \frac{3}{\sqrt{2}} - \frac{1}{\sqrt{2}} = \sqrt{2}$$

$$v \cdot \hat{u}_2 = -\frac{3}{\sqrt{3}} - \frac{1}{\sqrt{3}} + \frac{2}{\sqrt{3}} = -\frac{2}{\sqrt{3}}$$

$$v \cdot \hat{u}_3 = \frac{3}{\sqrt{7}} + \frac{5}{\sqrt{7}} + \frac{1}{\sqrt{7}} + \frac{4}{\sqrt{7}} = \frac{11}{\sqrt{7}}$$

$$v \cdot \hat{u}_4 = \frac{3}{\sqrt{42}} - \frac{30}{\sqrt{42}} + \frac{1}{\sqrt{42}} + \frac{4}{\sqrt{42}} = -\frac{22}{\sqrt{42}},$$

and so

$$v = \sqrt{2}\hat{u}_1 - \frac{2}{\sqrt{3}}\hat{u}_2 + \frac{11}{\sqrt{7}}\hat{u}_3 - \frac{22}{\sqrt{42}}\hat{u}_4.$$

There's still work to be done, but it is comparatively simpler than solving the corresponding system of equations.

3.2 Orthogonal Projection

In Exercise **??**, we saw that Fourier expansion theorem gives us an efficient way of testing whether or not a given vector belongs to the span of an orthogonal set. When the answer is "no", the quantity we compute while testing turns out to be very useful: it gives the orthogonal projection of that vector onto the span of our orthogonal set. This turns out to be exactly the ingredient needed to solve certain minimum distance problems.

3.2.1 The Gram-Schmidt Procedure

You may recall the following from elementary linear algebra, or vector calculus. Given an nonzero vector u and a vector v, the projection of v onto u is given by

$$\text{proj}_u\, v = \left(\frac{v \cdot u}{\|u\|^2}\right) u.$$

Note that this looks just like one of the terms in Fourier expansion theorem. Recall also that the vector $v - \text{proj}_u\, v$ is orthogonal to u. Our next result is a generalization of this observation.

Theorem 3.2.1 Orthogonal Lemma.

Let $\{v_1, v_2, \ldots, v_m\}$ be an orthogonal set of vectors in $\mathbb{R}^n$, and let x be any vector in $\mathbb{R}^n$. Define the vector v_{m+1} by

$$v_{m+1} = x - \left(\frac{x \cdot v_1}{\|v_1\|^2}v_1 + \cdots + \frac{x \cdot v_m}{\|v_m\|^2}v_m\right).$$

> **Then:**
>
> 1. $\mathbf{v}_{m+1} \cdot \mathbf{v}_i = 0$ for each $i = 1, \ldots, m$.
>
> 2. If $\mathbf{x} \notin span\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$, then $\mathbf{v}_{m+1} \neq \mathbf{0}$, and therefore, $\{\mathbf{v}_1, \ldots, \mathbf{v}_m, \mathbf{v}_{m+1}\}$ is an orthogonal set.

Proof.

1. *For any $i = 1, \ldots m$, we have*

$$\mathbf{v}_{m+1} \cdot \mathbf{v}_i = \mathbf{x} \cdot \mathbf{v}_i - \frac{\mathbf{x} \cdot \mathbf{v}_i}{\|\mathbf{v}_i\|^2}(\mathbf{v}_i \cdot \mathbf{v}_i) = 0,$$

since $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ for $i \neq j$.

2. *It follows from the Fourier expansion theorem that $\mathbf{v}_{m+1} = \mathbf{0}$ if and only if $\mathbf{x} \in span\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$, and the fact that $\{\mathbf{v}_1, \ldots, \mathbf{v}_m, \mathbf{v}_{m+1}\}$ is an orthogonal set then follows from the first part.*

∎

It follows from the Orthogonal Lemma that for any subspace $U \subseteq \mathbb{R}^n$, any set of orthogonal vectors in U can be extended to an orthogonal basis of U. Since any set containing a single nonzero vector is orthogonal, it follows that every subspace has an orthogonal basis. (If $U = \{\mathbf{0}\}$, we consider the empty basis to be orthogonal.)

The procedure for creating an orthogonal basis is clear. Start with a single nonzero vector $\mathbf{v}_1 \in U$. If $U \neq span\{\mathbf{v}_1\}$, choose a vector $\mathbf{x}_1 \in U$ with $\mathbf{x}_1 \notin span\{\mathbf{v}_1\}$. The Orthogonal Lemma then provides us with a vector

$$\mathbf{v}_2 = \mathbf{x}_1 - \frac{\mathbf{x}_1 \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|^2}\mathbf{v}_1$$

such that $\{\mathbf{v}_1, \mathbf{v}_2\}$ is orthogonal. If $U = span\{\mathbf{v}_1, \mathbf{v}_2\}$, we're done. Otherwise, we repeat the process, choosing $\mathbf{x}_2 \notin span\{\mathbf{v}_1, \mathbf{v}_2\}$, and then using the Orthogonal Lemma to obtain $\mathbf{v}_3$, and so on, until an orthogonal basis is obtained.

With one minor modification, the above procedure provides us with a major result. Suppose U is a subspace of $\mathbb{R}^n$, and start with any basis $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of U. By choosing our $\mathbf{x}_i$ in the procedure above to be these basis vectors, we obtain the Gram-Schmidt algorithm for constructing an orthogonal basis.

> ### Theorem 3.2.2 Gram-Schmidt Orthonormalization Algorithm.
>
> Let U be a subspace of $\mathbb{R}^n$, and let $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a basis of U. Define vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ in U as follows:
>
> $$\mathbf{v}_1 = \mathbf{x}_1$$
>
> $$\mathbf{v}_2 = \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|^2}\mathbf{v}_1$$
>
> $$\mathbf{v}_3 = \mathbf{x}_3 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|^2}\mathbf{v}_1 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_2}{\|\mathbf{v}_2\|^2}\mathbf{v}_2$$
>
> $$\vdots$$
>
> $$\mathbf{v}_m = \mathbf{x}_m - \frac{\mathbf{x}_m \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|^2}\mathbf{v}_1 - \cdots - \frac{\mathbf{x}_m \cdot \mathbf{v}_{m-1}}{\|\mathbf{v}_{m-1}\|^2}\mathbf{v}_{m-1}.$$
>
> Then $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is an orthogonal basis for U. Moreover, for each $k = 1, 2, \ldots, m$, we have
>
> $$span\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} = span\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}.$$

Of course, once we've used Gram-Schmidt to find an orthogonal basis, we can normalize each vector to get an orthonormal basis. The Gram-Schmidt algorithm is ideal when we know how to find a basis for a subspace, but we need to know an orthogonal basis. For example, suppose we want an orthonormal basis for the nullspace of the matrix

$$A = \begin{bmatrix} 2 & -1 & 3 & 0 & 5 \\ 0 & 2 & -3 & 1 & 4 \\ -4 & 2 & -6 & 0 & -10 \\ 2 & 1 & 0 & 1 & 9 \end{bmatrix}.$$

First, we find any basis for the nullspace.

```
from sympy import Matrix, init_printing
init_printing()
A = Matrix([[2,-1,3,0,5],
            [0,2,-3,1,4],
            [-4,2,-6,0,-10],
            [2,1,0,1,9]])
A.nullspace()
```

\[\left[\bbm -\frac34\\ \frac32\\1\\0\\0\ebm, \bbm
    -\frac14\\ -\frac12\\0\\1\\0\ebm, \bbm
    -\frac72\\-2\\0\\0\\1\ebm\right]\]

Let's make that basis look a little nicer by using some scalar multiplication to clear fractions.

$$B = \left\{ \mathbf{x}_1 = \begin{bmatrix} 3 \\ -6 \\ -4 \\ 0 \\ 0 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 0 \\ -4 \\ 0 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 7 \\ 4 \\ 0 \\ 0 \\ -2 \end{bmatrix} \right\}$$

Definitely not an orthogonal basis. So we take $\mathbf{v}_1 = \mathbf{x}_1$*, and*

$$\mathbf{v}_2 = \mathbf{x}_2 - \left(\frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|^2} \right) \mathbf{v}_1$$

$$= \begin{bmatrix} 1 \\ 2 \\ 0 \\ -4 \\ 0 \end{bmatrix} - \frac{-9}{61} \begin{bmatrix} 3 \\ -6 \\ -4 \\ -0 \\ 0 \end{bmatrix},$$

which equals something I'm not sure I want to try to simplify. Finally, we find

$$\mathbf{v}_3 = \mathbf{x}_3 - \left(\frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\|\mathbf{v}_1\|^2} \right) \mathbf{v}_1 - \left(\frac{\mathbf{x}_3 \cdot \mathbf{v}_2}{\|\mathbf{v}_2\|^2} \right) \mathbf{v}_2.$$

And now you probably get about five minutes into the fractions and say something that shouldn't appear in print. This sounds like a job for the computer.

```
from sympy import GramSchmidt
B = A.nullspace()
GramSchmidt(B)
```

\[\left[\bbm -\frac34\\ \frac32\\1\\0\\0\ebm, \bbm
    -\frac{22}{61}\\-\frac{17}{61}\\ \frac{9}{61}\\1\\0\ebm,
    \bbm

```
-\frac{76}{25}\\-\frac{36}{25}\\-\frac{3}{25}\\-\frac{37}{25}\\1\ebm\right]\]
```

Oh wait, you wanted that normalized? Turns out the `GramSchmidt` function has an optional argument of true or false. The default is false, which is to not normalize. Setting it to true gives an orthonormal basis:

```
GramSchmidt(B,true)
```

```
\[\left[\bbm -\frac{3\sqrt{61}}{61}\\
    \frac{6\sqrt{61}}{61}\\ \frac{4\sqrt{61}}{61}\\0\\0\ebm,
\bbm -\frac{22\sqrt{183}}{915}\\-\frac{17\sqrt{183}}{915}\\
    \frac{3\sqrt{183}}{305}\\ \frac{\sqrt{183}}{15}\\0\ebm,
\bbm
    -\frac{76\sqrt{3}}{165}\\-\frac{12\sqrt{3}}{55}\\-\frac{\sqrt{3}}{55}\\-\frac{37\sqrt{3}}{165}\\
    \frac{5\sqrt{3}}{33}\ebm\right]\]
```

OK, so that's nice, and fairly intimidating looking. Did it work? We can specify the vectors in our list by giving their positions, which are 0, 1, and 2, respectively.

```
L=GramSchmidt(B)
L[0],L[1],L[2]
```

```
\[\left(\bbm -\frac34\\ \frac32\\1\\0\\0\ebm, \bbm
    -\frac{22}{61}\\-\frac{17}{61}\\ \frac{9}{61}\\1\\0\ebm,
    \bbm
    -\frac{76}{25}\\-\frac{36}{25}\\-\frac{3}{25}\\-\frac{37}{25}\\1\ebm\right)\]
```

Let's compute dot products:

```
L[0].dot(L[1]),L[1].dot(L[2]),L[0].dot(L[2])
```

```
\[(0,0,0)\]
```

Let's also confirm that these are indeed in the nullspace.

```
A*L[0],A*L[1],A*L[2]
```

```
\[\left(\bbm 0\\0\\0\\0\ebm, \bbm 0\\0\\0\\0\ebm, \bbm
    0\\0\\0\\0\ebm\right)\]
```

Boom. Let's try another example. This time we'll keep the vectors a little smaller in case you want to try it by hand.

Exercise 3.2.3

Confirm that the set $B = \{(1, -2, 1), (3, 0, -2), (-1, 1, 2)\}$ is a basis for $\mathbb{R}^3$, and use the Gram-Schmidt Orthonormalization Algorithm to find an orthonormal basis.

3.2.2 Projections

We hinted above that the calculations we've been doing have a lot to do with projection. Since any single nonzero vector forms an orthogonal basis for its span, the

You'll notice that I used $6\mathbf{v}_2$ rather than $\mathbf{v}_2$ in the calculation of $\mathbf{v}_3$. This lets me avoid fractions (momentarily), and doesn't affect the answer, since for any

nonzero scalar c,

$$\left(\frac{c\mathbf{v}\cdot\mathbf{x}}{\|c\mathbf{v}\|^2}\right)(c\mathbf{v})$$

$$= \left(\frac{c(\mathbf{v}\cdot\mathbf{x})}{c^2\|\mathbf{v}\|^2}\right)(c\mathbf{v})$$

$$= \left(\frac{\mathbf{v}\cdot\mathbf{x}}{\|\mathbf{v}^2\|}\right)\mathbf{v}.$$

projection

$$\text{proj}_{\boldsymbol{u}}\, \boldsymbol{v} = \left(\frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|^2} \right) \boldsymbol{u}$$

can be viewed as the orthogonal projection of the vector $\boldsymbol{v}$*, not onto the vector* $\boldsymbol{u}$*, but onto the subspace* $\text{span}\{\boldsymbol{u}\}$*. This is, after all, how we viewed projections in elementary linear algebra: we drop the perpendicular from the tip of* $\boldsymbol{v}$ *onto the line in the direction of* $\boldsymbol{u}$*.*

Now that we know how to define an orthogonal basis for a subspace, we can define orthogonal projection onto subspaces of dimension greater than one.

Definition 3.2.4

Let U *be a subspace of* $\mathbb{R}^n$ *with orthogonal basis* $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\}$*. For any vector* $\boldsymbol{v} \in \mathbb{R}^n$*, we define the **orthogonal projection** of* $\boldsymbol{v}$ *onto* U *by*

$$\text{proj}_U\, \boldsymbol{v} = \sum_{i=1}^{k} \left(\frac{\boldsymbol{u}_i \cdot \boldsymbol{v}}{\|\boldsymbol{u}_i\|^2} \right) \boldsymbol{u}_i.$$

Note that $\text{proj}_U\, \boldsymbol{v}$ *is indeed an element of* U*, since it's a linear combination of its basis vectors. In the case of the trivial subspace* $U = \{\boldsymbol{0}\}$*, we define orthogonal projection of any vector to be* $\boldsymbol{0}$*, since really, what other choice do we have? (This case isn't really of any interest, we just like being thorough.)*

Let's see how this might be put to use in a classic problem: finding the distance from a point to a plane.

Example 3.2.5

Find the distance from the point $(3, 1, -2)$ *to the plane* P *defined by* $x - 2y + 4z = 0$*.*

Solution 1 *(Using projection onto a normal vector). In an elementary linear algebra (or calculus) course, we would solve this problem as follows. First, we would need two vectors parallel to the plane. If* $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ *lies in the plane, then* $x - 2y + 4z = 0$*, so* $x = 2y - 4z$*, and*

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2y - 4z \\ y \\ z \end{bmatrix} = y \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} + z \begin{bmatrix} -4 \\ 0 \\ 1 \end{bmatrix},$$

so $\boldsymbol{u} = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$ *and* $\boldsymbol{v} \begin{bmatrix} -4 \\ 0 \\ 1 \end{bmatrix}$ *are parallel to the plane. We then compute the normal vector*

$$\boldsymbol{n} = \boldsymbol{u} \times \boldsymbol{v} = \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix},$$

and compute the projection of the position vector $\boldsymbol{p} = \begin{bmatrix} 3, 1, -2 \end{bmatrix}$ *for the point* $P = (3, 1, -2)$ *onto* $\boldsymbol{n}$*. This gives the vector*

$$\boldsymbol{x} = \left(\frac{\boldsymbol{p} \cdot \boldsymbol{n}}{\|\boldsymbol{n}\|^2} \right) \boldsymbol{n} = \frac{-7}{21} \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix} = \begin{bmatrix} -1/3 \\ 2/3 \\ -4/3 \end{bmatrix}.$$

One limitation of this approach to projection is that we must project onto a subspace. Given a plane like $x - 2y + 4z = 4$*, we would need to modify our approach. One way to do it would be to find a point on the plane, and then try to translate everything to the origin. It's interesting to think about how this might be accomplished (in particular, in what direction would the translation have to be performed?) but someone external to the questions we're interested in here.*

Now, this vector is parallel to **n**, so it's perpendicular to the plane. Subtracting it from **p** gives a vector parallel to the plane, and this is the position vector for the point we seek.

$$\boldsymbol{q} = \boldsymbol{p} - \boldsymbol{x} = \begin{bmatrix} 3 \\ 1 \\ -2 \end{bmatrix} - \begin{bmatrix} -1/3 \\ -2/3 \\ -4/3 \end{bmatrix} = \begin{bmatrix} 10/3 \\ 1/3 \\ -2/3 \end{bmatrix}$$

so the closest point is $Q = \left(\frac{10}{3}, \frac{1}{3}, -\frac{2}{3} \right)$. We weren't asked for it, but note that if we wanted the distance from the point P to the plane, this is given by $\|\boldsymbol{x}\| = \frac{1}{3}\sqrt{21}$.

Solution 2 (Using orthogonal projection). Let's solve the same problem using orthogonal projection. First, we have to deal with the fact that the vectors **u** and **v** are probably not orthogonal. To get around this, we replace **v** with

$$\boldsymbol{w} = \boldsymbol{v} - \left(\frac{\boldsymbol{v} \cdot \boldsymbol{u}}{\|\boldsymbol{u}\|^2} \right) \boldsymbol{u} = \begin{bmatrix} -4 \\ 0 \\ 1 \end{bmatrix} + \frac{8}{5} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -4/5 \\ 8/5 \\ 1 \end{bmatrix}.$$

We now set

$$\boldsymbol{q} = \left(\frac{\boldsymbol{p} \cdot \boldsymbol{u}}{\|\boldsymbol{u}\|^2} \right) \boldsymbol{u} - \left(\frac{\boldsymbol{p} \cdot \boldsymbol{w}}{\|\boldsymbol{w}\|^2} \right) \boldsymbol{w}$$

$$= \frac{7}{5} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} + \frac{-14}{105} \begin{bmatrix} -4 \\ 8 \\ 5 \end{bmatrix}$$

$$= \begin{bmatrix} 10/3 \\ 1/3 \\ -2/3 \end{bmatrix}.$$

Lo and behold, we get the same answer as before.

The only problem with Definition **??** is that it appears to depend on the choice of orthogonal basis. To see that it doesn't, we need one more definition.

Definition 3.2.6

For any subspace U of $\mathbb{R}^n$, we define the **orthogonal complement** of U, denoted $U^\perp$, by

$$U^\perp = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \cdot \boldsymbol{y} = 0 \text{ for all } \boldsymbol{y} \in U\}.$$

The term "complement" comes from terminology we mentioned early on, but didn't spend much time on. Theorem **??** told us that for any subspace U of a vector space V, it is possible to construct another subspace W of V such that $V = U \oplus W$. The subspace W is known as a complement of U. A complement is not unique, but the orthogonal complement is. As you might guess from the name, $U^\perp$ is also a subspace of $\mathbb{R}^n$.

Exercise 3.2.7

Show that $U^\perp$ is a subspace of $\mathbb{R}^n$.

> **Theorem 3.2.8 Projection Theorem.**
>
> Let U be a subspace of $\mathbb{R}^n$, let x be any vector in $\mathbb{R}^n$, and let $p = proj_U\, x$. Then:
>
> 1. $p \in U$, and $x - p \in U^{\perp}$.
>
> 2. p is the closest vector in U to the vector x, in the sense that the distance $d(p, x)$ is minimal among all vectors in U. That is, for all $u \neq p \in U$, we have
>
> $$\|x - p\| < \|x - u\|.$$

Exercise 3.2.9

Show that $U \cap U^{\perp} = \{0\}$. Use this fact to show that Definition **??** does not depend on the choice orthogonal basis.

Hint. Suppose we find vectors p and p' using basis B and B'. Note that $p - p' \in U$, but also that

$$p - p' = (p - x) - (p' - x)$$

Now use Theorem **??**

Finally, we note one more useful fact. The process of sending a vector to its orthogonal projection defines an operator on $\mathbb{R}^n$, and yes, it's linear.

> **Theorem 3.2.10**
>
> Let U be a subspace of $\mathbb{R}^n$, and define a function $P_U : \mathbb{R}^n \to \mathbb{R}^n$ by
>
> $$P_U(x) = proj_U\, x \text{ for any } x \in \mathbb{R}^n.$$
>
> Then T is a linear operator such that $U = im\, P_U$ and $U^{\perp} = ker\, P_U$.

Note: it follows from this result and the Dimension Theorem that

$$dim\, U + dim\, U^{\perp} = n,$$

and since $U \cap U^{\perp} = \{0\}$, $U^{\perp}$ is indeed a complement of U in the sense introduced in Theorem **??**. It's also fairly easy to see that $dim\, U + dim\, U^{\perp} = n$ directly. If $w \in U^{\perp}$, and $\{u_1, \ldots, u_k\}$ is a basis for U, then we have

$$w \cdot u_1 = 0, \ldots, w \cdot u_k = 0,$$

and for an unknown w, this is simply a homogeneous system of k equations with n variables. Moreover, they are independent equations, since the u_i form a basis. We thus expect $n - k$ free parameters in the general solution.

Exercise 3.2.11

Let $U = \{(a - b + 3c, 2a + b, 3c, 4a - b + 3c, a - 4c) \mid a, b, c \in \mathbb{R}\} \subseteq \mathbb{R}^5$. Determine a basis for $U^{\perp}$.

3.3 Worksheet: dual basis.

Let V be a vector space over $\mathbb{R}$. (That is, scalars are real numbers, rather than, say, complex.) A linear transformation $\phi : V \to \mathbb{R}$ is called a **linear functional**.

Here are some examples of linear functionals:

- The map $\phi : \mathbb{R}^3 \to \mathbb{R}$ given by $\phi(x, y, z) = 3x - 2y + 5z$.

- The evaluation map $ev_a : P_n(\mathbb{R}) \to \mathbb{R}$ given by $ev_a(p) = p(a)$. (For example, $ev_2(3 - 4x + 5x^2) = 2 - 4(2) + 5(2^2) = 14$.)

- The map $\phi : \mathcal{C}[a, b] \to \mathbb{R}$ given by $\phi(f) = \int_a^b f(x)\,dx$, where $\mathcal{C}[a, b]$ denotes the space of all continuous functions on $[a, b]$.

Note that for any vector spaces V, W, the set $\mathcal{L}(V, W)$ of linear transformations from V to W is itself a vector space, if we define

$$(S + T)(v) = S(v) + T(v), \quad \text{and} \quad (kT)(v) = k(T(v)).$$

In particular, given a vector space V, we denote the set of all linear functionals on V by $V^* = \mathcal{L}(V, \mathbb{R})$, and call this the dual space of V.

We make the following observations:

- If $\dim V = n$ and $\dim W = m$, then $\mathcal{L}(V, W)$ is isomorphic to the space M_{mn} of $m \times n$ matrices, so it has dimension mn.

- Since $\dim \mathbb{R} = 1$, if V is finite-dimensional, then $V^* = \mathcal{L}(V, \mathbb{R})$ has dimension $1n = n$.

- Since $\dim V^* = \dim V$, V and V^* are isomorphic.

Here is a basic example that is intended as a guide to your intuition regarding dual spaces. Take $V = \mathbb{R}^3$. Given any $v \in V$, define a map $\phi_v : V \to \mathbb{R}$ by $\phi_v(w) = v \cdot w$ (the usual dot product).

One way to think about this: if we write $v \in V$ as a column vector $\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$, then we can identify ϕ_v with v^T, where the action is via multiplication:

$$\phi_v(w) = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = v_1 w_1 + v_2 w_2 + v_3 w_3.$$

It turns out that this example can be generalized, but the definition of ϕ_v involves the dot product, which is particular to $\mathbb{R}^n$.

There is a generalization of the dot product, known as an inner product. (See Chapter 10 of Nicholson, for example.) On any inner product space, we can associate each vector $v \in V$ to a linear functional ϕ_v using the procedure above.

Another way to work concretely with dual vectors (without the need for inner products) is to define things in terms of a basis.

Given a basis $\{v_1, v_2, \ldots, v_n\}$ of V, we define the corresponding **dual basis** $\{\phi_1, \phi_2, \ldots, \phi_n\}$ of V^* by

$$\phi_i(v_j) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}.$$

Note that each ϕ_j is well-defined, since any linear transformation can be defined by giving its values on a basis.

For the standard basis on $\mathbb{R}^n$, note that the corresponding dual basis functionals are given by

$$\phi_j(x_1, x_2, \ldots, x_n) = x_j.$$

That is, these are the coordinate functions on $\mathbb{R}^n$.

1.   *Show that the dual basis is indeed a basis for V^*.*

Next, let V and W be vector spaces, and let $T : V \to W$ be a linear transformation. For any such T, we can define the dual map $T^* : W^* \to V^*$ by $T^*(\phi) = \phi \circ T$ for each $\phi \in W^*$.

2.   *Confirm that (a) $T^*(\phi)$ does indeed define an element of V^*; that is, a linear map from V to $\mathbb{R}$, and (b) that T^* is linear.*

3.   *Let $V = P(\mathbb{R})$ be the space of all polynomials, and let $D : V \to V$ be the derivative transformation $D(p(x)) = p'(x)$. Let $\phi : V \to \mathbb{R}$ be the linear functional defined by $\phi(p(x)) = \int_0^1 p(x)\, dx$.*
         What is the linear functional $D^*(\phi)$?

4.   *Show that dual maps satisfy the following properties: for any $S, T \in \mathcal{L}(V, W)$ and $k \in \mathbb{R}$,*

   (a) $(S + T)^* = S^* + T^*$

   (b) $(kS)^* = kS^*$

   (c) $(ST)^* = T^*S^*$

   In item Item 3.3.4.c, assume $S \in \mathcal{L}(V, W)$ and $T \in \mathcal{L}(U, V)$. (Reminder: the notation ST is sometimes referred to as the "product" of S and T, in analogy with matrices, but actually represents the composition $S \circ T$.)

   *We have one topic remaining in relation to dual spaces: determining the kernel and image of a dual map T^* (in terms of the kernel and image of T). Let V be a vector space, and let U be a subspace of V. Any such subspace determines an important subspace of V^*: the **annihilator** of U, denoted by U^0 and defined by*

$$U^0 = \{\phi \in V^* \mid \phi(u) = 0 \text{ for all } u \in U\}.$$

5.   *Determine a basis (in terms of the standard dual basis for $(\mathbb{R}^4)^*$) for the annihilator U^0 of the subspace $U \subseteq \mathbb{R}^4$ given by*

$$U = \{(2a + b, 3b, a, a - 2b) \mid a, b \in \mathbb{R}\}.$$

Here is a fun theorem about annihilators that I won't ask you to prove.

Theorem 3.3.1

Let V be a finite dimensional vector space. For any subspace U of V,

$$\dim U + \dim U^0 = \dim V.$$

Here's an outline of the proof. For any subspace $U \subseteq V$, we can define the inclusion map $i : U \to V$, given by $i(u) = u$. (This is not the identity on V since it's only defined on U. In particular, it is not onto unless $U = V$, although it is clearly one-to-one.)

Then i^* is a map from V^* to U^*. Moreover, note that for any $\phi \in V^*$, $i^*(\phi) \in U^*$ satisfies, for any $u \in U$,

$$i^*(\phi)(u) = \phi(i(u)) = \phi(u).$$

Thus, $\phi \in \ker i^*$ if and only if $i^*(\phi) = 0$, which is if and only if $\phi(u) = 0$ for all $u \in U$, which is if and only if $\phi \in U^0$. Therefore, $\ker i^* = U^0$.

By the dimension theorem, we have:

$$\dim V^* = \dim \ker i^* + \dim \operatorname{im} i^*.$$

With a bit of work, one can show that $\operatorname{im} i^* = U^*$, and we get the result from the fact that $\dim V^* = \dim V$ and $\dim U^* = \dim U$.

There are a number of interesting results of this flavour. For example, one can show that a map T is injective if and only if T^* is surjective, and vice-versa.

One final, optional task: return to the example of $\mathbb{R}^n$, viewed as column vectors, and consider a matrix transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$ given by $T_A(\vec{x}) = A\vec{x}$ as usual. Viewing $(\mathbb{R}^n)^*$ as row vectors, convince yourself that $(T_A)^* = T_{A^T}$; that is, that what we've really been talking about all along is just the transpose of a matrix!

Chapter 4

Diagonalization

In this chapter we look at the diagonalization problem for real symmetric matrices. You probably saw how to compute eigenvalues and eigenvectors in your elementary linear algebra course. You may have also seen that in some cases, the number of independent eigenvectors associated to an $n \times n$ matrix A is n, in which case it is possible to "diagonalize" A. In other cases, we don't get "enough" eigenvectors for diagonalization.

In the first part of this section, we review some basic facts about eigenvalues and eigenvectors. We will then move on to look at the special case of symmetric matrices, where we will see that it is always possible to diagonalize, and moreover, that it is possible to do so using an orthonormal basis of eigenvectors.

4.1 Eigenvalues and Eigenvectors

> **Definition 4.1.1**
>
> Let A be an $n \times n$ matrix. A number λ is called an **eigenvalue** of A if there exists a nonzero vector $\mathbf{x}$ such that
>
> $$A\mathbf{x} = \lambda\mathbf{x}.$$
>
> Any such vector $\mathbf{x}$ is called an **eigenvector** associated to the eigenvalue λ.

Note that eigenvalues and eigenvectors can just as easily be defined for a general linear operator $T : V \to V$. In this context, and eigenvector $\mathbf{x}$ is sometimes referred to as a characteristic vector (or characteristic direction) for T, since the property $T(\mathbf{x}) = \lambda\mathbf{x}$ simply states that the transformed vector $T(\mathbf{x})$ is parallel to the original vector $\mathbf{x}$. Some linear algebra textbooks that focus more on general linear transformations frame this topic in the context of invariant subspaces for a linear operator.

A subspace $U \subseteq V$ is invariant with respect to T if $T(\mathbf{u}) \in U$ for all $\mathbf{u} \in U$. Note that if $\mathbf{x}$ is an eigenvector of T, then span$\{\mathbf{x}\}$ is an invariant subspace. To see this, note that if $T(\mathbf{x}) = \lambda\mathbf{x}$ and $\mathbf{y} = k\mathbf{x}$, then

$$T(\mathbf{y}) = T(k\mathbf{x}) = kT(\mathbf{x}) = k(\lambda\mathbf{x}) = \lambda(k\mathbf{x}) = \lambda\mathbf{y}.$$

Note that if $\mathbf{x}$ is an eigenvector of the matrix A, then we have

$$(A - \lambda I_n)\mathbf{x} = \mathbf{0}, \qquad\qquad (4.1.1)$$

where I_n denotes the $n \times n$ identity matrix. Thus, if λ is an eigenvalue of A, any corresponding eigenvector is an element of null$(A - \lambda I_n)$.

> **Definition 4.1.2**
>
> For any real number λ and matrix A, we define the **eigenspace** $E_\lambda(A)$ by
>
> $$E_\lambda(A) = \text{null}(A - \lambda I_n).$$

Note that $E_\lambda(A)$ can be defined for any real number λ, whether or not λ is an eigenvalue. However, the eigenvalues of A are distinguished by the property that there is a nonzero solution to (**??**). Furthermore, we know that (**??**) can only have nontrivial solutions if the matrix $A - \lambda I_n$ is not invertible. We also know that $A - \lambda I_n$ is non-invertible if and only if $\det(A - \lambda I_n) = 0$. This gives us the following theorem.

Theorem 4.1.3

The following are equivalent for any $n \times n$ matrix A and real number λ:

1. λ is an eigenvalue of A.

2. $E_\lambda(A) \neq \{\mathbf{0}\}$

3. $\det(A - \lambda I_n) = 0$

The polynomial $p_A(x) = \det(xI_n - A)$ is called the **characteristic polynomial** of A. (Note that $\det(xI_n - A) = (-1)^n \det(A - xI_n)$. We choose this order so that the coefficient of x^n is always 1.) The equation

$$\det(xI_n - A) = 0 \tag{4.1.2}$$

is called the **characteristic equation** of A. The solutions to this equation are precisely the eigenvalues of A.

Recall that a matrix B is said to be **similar** to a matrix A if there exists an invertible matrix P such that $B = P^{-1}AP$. Much of what follows concerns the question of whether or not a given $n \times n$ matrix A is **diagonalizable**.

Definition 4.1.4

An $n \times n$ matrix A is said to be **diagonalizable** if A is similar to a diagonal matrix.

The following results will frequently be useful.

Theorem 4.1.5

The relation $A \sim B$ if and only if A is similar to B is an equivalence relation. Moreover, if $A \sim B$, then:

- $\det A = \det B$

- $\operatorname{tr} A = \operatorname{tr} B$

- $c_A(x) = c_B(x)$

In other words, A and B have the same determinant, trace, and characteristic polynomial (and thus, the same eigenvalues).

Proof. The first two follow directly from properties of the determinant and trace. For the last, note that if $B = P^{-1}AP$, then

$$P^{-1}(xI_n - A)P = P^{-1}(xI_n)P - P^{-1}AP = xI_nB,$$

so $xI_n - B \sim xI_n - A$, and therefore $\det(xI_n - B) = \det(xI_n - A)$. ∎

Example 4.1.6

Determine the eigenvalues and eigenvectors of $A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$.

Solution. We begin with the characteristic polynomial. We have

$$det(xI_n - A) = det \begin{bmatrix} x & -1 & -1 \\ -1 & x & -1 \\ -1 & -1 & x \end{bmatrix}$$

$$= x \begin{vmatrix} x & -1 \\ -1 & x \end{vmatrix} + 1 \begin{vmatrix} -1 & -1 \\ -1 & x \end{vmatrix} - 1 \begin{vmatrix} -1 & x \\ -1 & -1 \end{vmatrix}$$

$$= x(x^2 - 1) + (-x - 1) - (1 + x)$$

$$x(x - 1)(x + 1) - 2(x + 1)$$

$$(x + 1)[x^2 - x - 2]$$

$$(x + 1)^2(x - 2).$$

The roots of the characteristic polynomial are our eigenvalues, so we have $\lambda_1 = -1$ and $\lambda_2 = 2$. Note that the first eigenvalue comes from a repeated root. This is typically where things get interesting. If an eigenvalue does not come from a repeated root, then there will only be one (independent) eigenvector that corresponds to it. (That is, dim $E_\lambda(A) = 1$.) If an eigenvalue is repeated, it could have more than one eigenvector, but this is not guaranteed.

We find that $A - (-1)I_n = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, which has reduced row-echelon

form $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Solving for the nullspace, we find that there are two independent eigenvectors:

$$x_{1,1} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \quad and \quad x_{1,2} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix},$$

so

$$E_{-1}(A) = span \left\{ \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right\}.$$

For the second eigenvector, we have $A - 2I = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}$, which

has reduced row-echelon form $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$. An eigenvector in this case is

given by

$$x_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

In general, if the characteristic polynomial can be factored as

$$p_A(x) = (x - \lambda)^m q(x),$$

*where $q(x)$ is not divisible by $x - \lambda$, then we say that λ is an eigenvalue of **multiplicity** m. A main result is the following.*

> ### Theorem 4.1.7
>
> **Let λ be an eigenvalue of A of multiplicity m. Then** *dim* $E_\lambda(A) \leq m$.

Some textbooks refer to the multiplicity m of an eigenvalue as the algebraic multiplicity of λ, and the number dim $E_\lambda(A)$ as the geometric multiplicity of λ.

To prove Theorem ?? we need the following lemma, from Section 5.5 of Nicholson's textbook.

> ### Lemma 4.1.8
>
> **Let $\{x_1, \ldots, x_k\}$ be a set of linearly independent eigenvectors of a matrix A, with corresponding eigenvalues $\lambda_1, \ldots, \lambda_k$ (not necessarily distinct). Extend this set to a basis $\{x_1, \ldots, x_k, x_{k+1}, \ldots, x_n\}$, and let $P = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$ be the matrix whose columns are the basis vectors. (Note that P is necessarily invertible.) Then**
>
> $$P^{-1}AP = \begin{bmatrix} diag(\lambda_1, \ldots, \lambda_k) & B \\ 0 & A_1 \end{bmatrix},$$
>
> **where B has size $k \times (n - k)$, and A_1 has size $(n - k) \times (n - k)$.**

Proof. *We have*

$$P^{-1}AP = P^{-1}A \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$
$$= \begin{bmatrix} (P^{-1}A)x_1 & \cdots & (P^{-1}A)x_n \end{bmatrix}.$$

For $1 \leq i \leq k$, we have

$$(P^{-1}A)(x_i) = P^{-1}(Ax_i) = P^{-1}(\lambda_i x_i) = \lambda_i(P^{-1}x_i).$$

But $P^{-1}x_i$ is the ith column of $P^{-1}P = I_n$, which proves the result. ∎

We can use Lemma ?? to prove that dim $E_\lambda(A) \leq m$ as follows. Suppose $\{x_1, \ldots, x_k\}$ is a basis for $E_\lambda(A)$. Then this is a linearly independent set of eigenvectors, so our lemma guarantees the existence of a matrix P such that

$$P^{-1}AP = \begin{bmatrix} \lambda I_k & B \\ 0 & A_1 \end{bmatrix}.$$

Let $\tilde{A} = P^{-1}AP$. On the one hand, since $\tilde{A} \sim A$, we have $c_A(x) = c_{\tilde{A}}(x)$. On the other hand,

$$det(xI_n - \tilde{A}) = det \begin{bmatrix} (x - \lambda)I_k & -B \\ 0 & xI_{n-k} - A_1 \end{bmatrix} = (x - \lambda)^k det(xI_{n-k} - A_1).$$

This shows that $c_A(x)$ is divisible by $(x - \lambda)^k$. Since m is the largest integer such that $c_A(x)$ is divisible by $(x - \lambda)^m$, we must have dim $E_\lambda(A) = k \leq m$.

Theorem ?? provides an initial criterion for diagonalization: if the dimension of each eigenspace $E_\lambda(A)$ is equal to the multiplicity of λ, then A is diagonalizable. The truth of this statement relies on one additional fact: any set of eigenvectors corresponding to distinct eigenvalues is linearly independent. The proof of this fact is a relatively straightforward proof by induction. It can be found in Section 5.5 of Nicholson for those who are interested. However, our focus for the remainder of the section will be on diagonalization of symmetric matrices, and soon we will see that for such

matrices, eigenvectors corresponding to different eigenvalues are, in fact, orthogonal.

4.2 Diagonalization of symmetric matrices

*Recall that an $n \times n$ matrix A is **symmetric** if $A^T = A$. Symmetry of A is equivalent to the following: for any vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,*

$$\mathbf{x} \cdot (A\mathbf{y}) = (A\mathbf{x}) \cdot \mathbf{y}.$$

To see that this is implied by the symmetry of A, note that

$$\mathbf{x} \cdot (A\mathbf{y}) = \mathbf{x}^T (A\mathbf{y}) = (\mathbf{x}^T A^T)\mathbf{y} = (A\mathbf{x})^T \mathbf{y} = (A\mathbf{x}) \cdot \mathbf{y}.$$

For inner product spaces, the above is taken as the definition of what it means for an operator to be symmetric.

> **Exercise 4.2.1**
>
> Prove that if $\mathbf{x} \cdot (A\mathbf{y}) = (A\mathbf{x}) \cdot \mathbf{y}$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, then A is symmetric.

A useful property of symmetric matrices, mentioned earlier, is that eigenvectors corresponding to distinct eigenvalues are orthogonal.

> **Theorem 4.2.2**
>
> **If A is a symmetric matrix, then eigenvectors corresponding to distinct eigenvalues are orthogonal.**

Proof. *To see this, suppose A is symmetric, and that we have*

$$A\mathbf{x}_1 = \lambda_1 \mathbf{x}_1 \quad \text{and } A\mathbf{x}_2 = \lambda_2 \mathbf{x}_2,$$

with $\mathbf{x}_1 \neq \mathbf{0}, \mathbf{x}_2 \neq \mathbf{0}$, and $\lambda_1 \neq \lambda_2$. We then have, since A is symmetric, and using the result above,

$$\lambda_1(\mathbf{x}_1 \cdot \mathbf{x}_2) = (\lambda_1 \mathbf{x}_1) \cdot \mathbf{x}_2 = (A\mathbf{x}_1) \cdot \mathbf{x}_2 = \mathbf{x}_1 \cdot (A\mathbf{x}_2) = \mathbf{x}_1(\lambda_2 \mathbf{x}_2) = \lambda_2(\mathbf{x}_1 \cdot \mathbf{x}_2).$$

It follows that $(\lambda_1 - \lambda_2)(\mathbf{x}_1 \cdot \mathbf{x}_2) = 0$, and since $\lambda_1 \neq \lambda_2$, we must have $\mathbf{x}_1 \cdot \mathbf{x}_2 = 0$. ∎

The procedure for diagonalizing a matrix is as follows: assuming that $\dim E_\lambda(A)$ is equal to the multiplicity of λ for each distinct eigenvalue λ, we find a basis for $E_\lambda(A)$. The union of the bases for each eigenspace is then a basis of eigenvectors for $\mathbb{R}^n$, and the matrix P whose columns are those eigenvectors will satisfy $P^{-1}AP = D$, where D is a diagonal matrix whose diagonal entries are the eigenvalues of A.

If A is symmetric, we know that eigenvectors from different eigenspaces will be orthogonal to each other. If we further choose an orthogonal basis of eigenvectors for each eigenspace (which is possible via the Gram-Schmidt procedure), then we can construct an orthogonal basis of eigenvectors for $\mathbb{R}^n$. Furthermore, if we normalize each vector, then we'll have an orthonormal basis. The matrix P whose columns consist of these orthonormal basis vectors has a name.

> **Definition 4.2.3**
>
> *A matrix P is called **orthogonal** if $P^T = P^{-1}$.*

> **Theorem 4.2.4**
>
> **A matrix P is orthogonal if and only if the columns of P form an ortho-normal basis for $\mathbb{R}^n$.**

A fun fact is that if the columns of P are orthonormal, then so are the rows. But this is not true if we ask for the columns to be merely orthogonal. For example, the columns of $A = \begin{bmatrix} 1 & 0 & 5 \\ -2 & 1 & 2 \\ 1 & 2 & -1 \end{bmatrix}$ are orthogonal, but (as you can check) the rows are not. But if we normalize the columns, we get

$$P = \begin{bmatrix} 1/\sqrt{6} & 0 & 1/\sqrt{30} \\ -2/\sqrt{6} & 1/\sqrt{5} & 2/\sqrt{30} \\ 1/\sqrt{6} & 2/\sqrt{5} & -1/\sqrt{30} \end{bmatrix},$$

which, as you can confirm, is an orthogonal matrix.

> **Definition 4.2.5**
>
> An $n \times n$ matrix A is said to be *orthogonally diagonalizable* if there exists an orthogonal matrix P such that $P^T A P$ is diagonal.

The above definition leads to the following result, also known as the *Principal Axes Theorem*.

> **Theorem 4.2.6 Real Spectral Theorem.**
>
> **The following are equivalent for a real $n \times n$ matrix A:**
>
> 1. **A is symmetric.**
>
> 2. **There is an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of A.**
>
> 3. **A is orthogonally diagonalizable.**

> **Exercise 4.2.7**
>
> Determine the eigenvalues of $A = \begin{bmatrix} 5 & -2 & -4 \\ -2 & 8 & -2 \\ -4 & -2 & 5 \end{bmatrix}$, and find an orthogonal matrix P such that $P^T A P$ is diagonal.

4.3 Quadratic forms

If you've done a couple of calculus courses, you've probably encountered conic sections, like the ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ or the parabola $\frac{y}{b} = \frac{x^2}{a^2}$. You might also recall that your instructor was careful to avoid conic sections with equations including "cross-terms" like xy. The reason for this is that sketching a conic section like $x^2 + 4xy + y^2 = 1$ requires the techniques of the previous section.

A basic fact about orthogonal matrices is that they preserve length. Indeed, for any vector $\mathbf{x}$ in $\mathbb{R}^n$ and any orthogonal matrix P,

$$\|P\mathbf{x}\|^2 = (P\mathbf{x}) \cdot (P\mathbf{x}) = (P\mathbf{x})^T (P\mathbf{x}) = (\mathbf{x}^T P^T)(P\mathbf{x}) = \mathbf{x}^T \mathbf{x} = \|\mathbf{x}\|^2,$$

since $P^T P = I_n$.

Note also that since $P^T P = I_n$ and $\det P^T = \det P$, we have

$$\det(P)^2 = \det(P^T P) = \det(I_n) = 1,$$

so $\det(P) = \pm 1$. If $\det P = 1$, we have what is called a special orthogonal matrix. In $\mathbb{R}^2$ or $\mathbb{R}^3$, multiplication by a special orthogonal matrix is simply a rotation. (If $\det P = -1$, there is also a reflection.)

We mentioned in the previous section that the Real Spectral Theorem is also re-ferred to as the principal axes theorem. The name comes from the fact that one way to interpret the orthogonal diagonalization of a symmetric matrix is that we are rotat-ing our coordinate system. The original coordinate axes are rotated to new coordinate axes, with respect to which the matrix A is diagonal. This will become more clear once we apply these ideas to the problem of conic sections mentioned above. First, a defi-nition.

Definition 4.3.1

A **quadratic form** on variables $x_1, x_2, \ldots, x_n$ is any expression of the form

$$q(x_1, \ldots, x_n) = \sum_{i \leq j} a_{ij} x_i x_j.$$

For example, $q_1(x, y) = 4x^2 - 4xy + 4y^2$ and $q_2(x, y, z) = 9x^2 - 4y^2 - 4xy - 2xz + z^2$ are quadratic forms. Note that each term in a quadratic form is of degree two. We omit linear terms, since these can be absorbed by completing the square. The important observation is that every quadratic form can be associated to a symmetric matrix. The diagonal entries are the coefficients a_{ii} appearing in Definition ??, while the off-diagonal entries are half the corresponding coefficients a_{ij}.

For example the two quadratic forms given above have the following associated matrices:

$$A_1 = \begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 9 & -2 & -1 \\ -2 & 4 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

The reason for this is that we can then write

$$q_1(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and

$$q_2(x, y, z) = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 9 & -2 & -1 \\ -2 & 4 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Of course, the reason for wanting to associate a symmetric matrix to a quadratic form is that it can be orthogonally diagonalized. Consider the matrix A_1.

```
from sympy import Matrix, init_printing, factor
init_printing()
A1 = Matrix(2,2,[4,-2,-2,4])
p = A1.charpoly().as_expr()
factor(p)
```

$$\[(\lambda-6)(\lambda-2)\]$$

We find distinct eigenvalues $\lambda_1 = 2$ and $\lambda_2 = 6$. Since A is symmetric, we know the corresponding eigenvectors will be orthogonal.

```
A1.eigenvects()
```

```
\[\left[\left(2,1,\bbm 1\\1\ebm\right),\left(6,1,\bbm
    -1\\1\ebm\right)\right]\]
```

The resulting orthogonal matrix is $P = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, and we find

$$P^T A P = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}, \text{ or } A = PDP^T,$$

where $D = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}$. If we define new variables y_1, y_2 by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = P^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

then we find that

$$\begin{aligned} \begin{bmatrix} x_1 & x_2 \end{bmatrix} A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= (\begin{bmatrix} x_1 & x_2 \end{bmatrix} P) D \left(P^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) \\ &= \begin{bmatrix} y_1 & y_2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= 2y_1^2 + 6y_2^2. \end{aligned}$$

Note that there is no longer any cross term.

Now, suppose we want to graph the conic $4x_1^2 - 4x_1x_2 + 4x_2^2 = 12$. By changing to the variables y_1, y_2 this becomes $2y_1^2 + 6y_2^2 = 12$, or $\frac{y_1^2}{6} + \frac{y_2^2}{2} = 1$. This is the standard from of an ellipse, but in terms of new variables. How do we graph it? Returning to the definition of our new variables, we find $y_1 = \frac{1}{\sqrt{2}}(x_1 + x_2)$ and $y_2 = \frac{1}{\sqrt{2}}(-x_1 + x_2)$. The y_1 axis should be the line $y_2 = 0$, or $x_1 = x_2$. (Note that this line points in the direction of the eigenvector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.) The y_2 axis should be the line $y_1 = 0$, or $x_1 = -x_2$, which is in the direction of the eigenvector $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$.

This lets us see that our new coordinate axes are simply a rotation (by $\pi/4$) of the old coordinate axes, and our conic section is, accordingly, an ellipse that has been rotated by the same angle.

4.4 Diagonalization of complex matrices

Recall that when we first defined vector spaces, we mentioned that a vector space can be defined over any field $\mathbb{F}$. To keep things simple, we've mostly assumed $\mathbb{F} = \mathbb{R}$. But most of the theorems and proofs we've encountered go through unchanged if we work over a general field. (This is not quite true: over a finite field things can get more complicated. For example, if $\mathbb{F} = \mathbb{Z}_2 = \{0, 1\}$, then we get weird results like $v + v = 0$, since $1 + 1 = 0$.)

In fact, if we replace $\mathbb{R}$ by , about the only thing we'd have to go back and change is the definition of the dot product. The reason for this is that although the complex numbers seem computationally more complicated, (which might mostly be because you don't use them often enough) they follow the exact same algebraic rules as the real numbers. In other words, the arithmetic might be different, but the algebra is the same. There is one key difference between the two fields: over the complex numbers, every polynomial can be factored. This is important if you're interested in finding eigenvalues.

4.4.1 Review of complex numbers

Let's quickly review some basic facts about complex numbers that are typically covered in an earlier course. First, we define the set of complex numbers by

$$= \{x + iy \mid x, y \in \mathbb{R}\},$$

where $i = \sqrt{-1}$. We have a bijection $\to \mathbb{R}^2$ given by $x + iy \mapsto (x, y)$; because of this, we often picture as the complex plane, with a "real" x axis, and an "imaginary" y axis.

Arithmetic with complex numbers is defined by

$$(x_1 + iy_1) + (x_2 + iy_2) = (x_1 + x_2) + i(y_1 + y_2)$$
$$(x_1 + iy_1)(x_2 + iy_2) = (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1).$$

The multiplication rule looks complicated, but it's really just "FOIL", along with the fact that $i^2 = -1$. Note that if $c = c + i0$ is real, we have $c(x + iy) = (cx) + i(cy)$, so that has the structure of a two dimensional vector space over $\mathbb{R}$ (isomorphic to $\mathbb{R}^2$).

*Subtraction is defined in the obvious way. Division is less obvious. To define division, it helps to first introduce the **complex conjugate**. Given a complex number $z = x + iy$, we define $\bar{z} = x - iy$. The importance of the conjugate is that we have the identity*

$$z\bar{z} = (x + iy)(x - iy) = x^2 + y^2.$$

*So $z\bar{z}$ is real, and non-negative. This lets us define the **modulus** of z by*

$$|z| = \sqrt{z\bar{z}} = \sqrt{x^2 + y^2}.$$

This gives a measure of the magnitude of a complex number, in the same way as the vector norm on $\mathbb{R}^2$.

Now, given $z = x + iy$ and $w = s + it$, we have

$$\frac{z}{w} = \frac{z\bar{w}}{w\bar{w}} = \frac{(x + iy)(s - it)}{s^2 + t^2} = \frac{xs - yt}{s^2 + t^2} + i\frac{xt + ys}{s^2 + t^2}.$$

And of course, we have $w\bar{w} \neq 0$ unless $w = 0$, and as usual, we don't divide by zero.

An important thing to keep in mind when working with complex numbers is that they follow the same algebraic rules as real numbers. For example, given a, b, z, w all complex, and $a \neq 0$, where $az + b = w$, if we want to solve for z, the answer is $z = \frac{1}{a}(w - b)$, as it would be in $\mathbb{R}$. The difference between $\mathbb{R}$ and only really materializes when we want to compute z, by plugging in values for a, b and w.

*One place where is computationally more complicated is finding powers and roots. For this, it is often more convenient to write our complex numbers in **polar form**. The key to the polar form for complex numbers is Euler's identity. For a unit complex number z (that is with $|z| = 1$), we can think of z as a point on the unit circle, and write*

$$z = \cos(\theta) + i\sin(\theta).$$

If $|z| = r$, we simply change the radius of our circle, so in general, $z = r(\cos(\theta) + i\sin(\theta))$. Euler's identity states that

$$\cos(\theta) + i\sin(\theta) = e^{i\theta}. \tag{4.4.1}$$

This idea of putting a complex number in an exponential function seems odd at first. If you take a course in complex variables, you'll get a better understanding of why this makes sense. But for now, we can take it as a convenient piece of notation. The reason it's convenient is that the rules for complex arithmetic turn out to align

quite nicely with properties of the exponential function. For example, de Moivre's Theorem states that

$$(\cos(\theta) + i\sin(\theta))^n = \cos(n\theta) + i\sin(n\theta).$$

This can be proved by induction (and the proof is not even all that bad), but it seems perfectly obvious in exponential notation:

$$(e^{i\theta})^n = e^{in\theta},$$

since you multiply exponents when you raise a power to a power.

Similarly, if we want to multiply two unit complex numbers, we have

$$\begin{aligned}(\cos\alpha + i\sin\alpha)(\cos\beta + i\sin\beta) &= (\cos\alpha\cos\beta - \sin\alpha\sin\beta)\\ &\quad + i(\sin\alpha\cos\beta + \cos\alpha\sin\beta)\\ &= \cos(\alpha + \beta) + i\sin(\alpha + \beta).\end{aligned}$$

But in exponential notation, this is simply

$$e^{i\alpha}e^{i\beta} = e^{i(\alpha+\beta)},$$

which makes sense, since when you multiply exponentials, you add the exponents.

Generally, problems involving addition and subtraction are best handled in "rectangular" $(x + iy)$ form, while problems involving multiplication and powers are best handled in polar form.

4.4.2 Complex vectors

A complex vector space is simply a vector space where the scalars are elements of $\mathbb{C}$ rather than $\mathbb{R}$. Examples include polynomials with complex coefficients, complex-valued functions, and $\mathbb{C}^n$, which is defined exactly how you think it should be. In fact, one way to obtain $\mathbb{C}^n$ is to start with the exact same standard basis we use for $\mathbb{R}^n$, and then take linear combinations using complex scalars.

We'll write elements of $\mathbb{C}^n$ as $\mathbf{z} = (z_1, z_2, \ldots, z_n)$. Notice that we've dropped the arrow notation for vectors in favour of a bold font. The reason is that we'll want to consider complex conjugates, and things get cluttered if we try to fit an arrow and a bar over our vector. The complex conjugate of $\mathbf{z}$ is given by

$$\bar{\mathbf{z}} = (\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_n).$$

The standard inner product on $\mathbb{C}^n$ looks a lot like the dot product on $\mathbb{R}^n$, with one important difference: we apply a complex conjugate to the second vector.

> **Definition 4.4.1**
>
> The **standard inner product** on $\mathbb{C}^n$ is defined as follows: given $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_n)$,
>
> $$\langle \mathbf{z}, \mathbf{w} \rangle = \mathbf{z} \cdot \bar{\mathbf{w}} = z_1\bar{w}_1 + z_2\bar{w}_2 + \cdots + z_n\bar{w}_n.$$

If $\mathbf{z}, \mathbf{w}$ are real, this is just the usual dot product. The reason for using the complex conjugate is to ensure that we still have a positive-definite inner product on $\mathbb{C}^n$:

$$\langle \mathbf{z}, \mathbf{z} \rangle = z_1\bar{z}_1 + z_2\bar{z}_2 + \cdots + z_n\bar{z}_n = |z_1|^2 + |z_2|^2 + \cdots + |z_n|^2,$$

which shows that $\langle \mathbf{z}, \mathbf{z} \rangle \geq 0$, and $\langle \mathbf{z}, \mathbf{z} \rangle = 0$ if and only if $\mathbf{z} = \mathbf{0}$.

> *Exercise 4.4.2*
>
> *Compute the dot product of* $\mathbf{z} = (2-i, 3i, 4+2i)$ *and* $\mathbf{w} = (3i, 4-5i, -2+2i)$.

This isn't hard to do by hand, but it's useful to know how to ask the computer to do it, too. Unfortunately, the dot product in SymPy does not include the complex conjugate. One likely reason for this is that while most mathematicians take the complex conjugate of the second vector, some mathematicians, and most physicists, put the conjugate on the first vector. So they may have decided to remain agnostic about this choice. We can manually apply the conjugate, using `Z.dot(W.H)`.

```
from sympy import Matrix,init_printing
init_printing()
Z = Matrix(3,1,[2-I,3*I,4+2*I])
W = Matrix(3,1,[3*I,4-5*I,-2+2*I])
Z, W, Z.dot(W.H)
```

$$\left(\begin{bmatrix} 2-i\\3i\\4+2i\end{bmatrix}, \begin{bmatrix} 3i\\4-5i\\-2+2i\end{bmatrix}, (-2-2i)(4+2i)-3i(2-i)+3i(4+5i)\right)$$

Again, you might want to wrap that last term in `simplify()` *(in which case you'll get* $-22 - 6i$ *for the dot product). Above, we saw that the complex inner product is designed to be positive definite, like the real inner product. The remaining properties of the complex inner product are given as follows.*

> **Theorem 4.4.3**
>
> **For any vectors $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ and any complex number α,**
>
> 1. $\langle \mathbf{z}_1 + \mathbf{z}_2, \mathbf{z}_3 \rangle = \langle \mathbf{z}_1, \mathbf{z}_3 \rangle + \langle z\mathbf{z}_2, \mathbf{z}_3 \rangle$ **and** $\langle \mathbf{z}_1, \mathbf{z}_2 + \mathbf{z}_3 \rangle = \langle \mathbf{z}_1, \mathbf{z}_2 \rangle + \langle z\mathbf{z}_1, \mathbf{z}_3 \rangle$.
>
> 2. $\langle \alpha\mathbf{z}_1, \mathbf{z}_2 \rangle = \alpha\langle \mathbf{z}_1, \mathbf{z}_2 \rangle$ **and** $\langle \mathbf{z}_1, \alpha\mathbf{z}_2 \rangle = \bar{\alpha}\langle \mathbf{z}_1, \mathbf{z}_2 \rangle$.
>
> 3. $\langle \mathbf{z}_2, \mathbf{z}_1 \rangle = \overline{\langle \mathbf{z}_1, \mathbf{z}_2 \rangle}$
>
> 4. $\langle \mathbf{z}_1, \mathbf{z}_1 \rangle \geq 0$, **and** $\langle \mathbf{z}_1, \mathbf{z}_1 \rangle = 0$ **if and only if** $\mathbf{z}_1 = \mathbf{0}$.

Proof.

1. *Using the distributive properties of matrix multiplication and the transpose,*

$$\begin{aligned} \langle \mathbf{z}_1 + \mathbf{z}_2, \mathbf{z}_3 \rangle &= (\mathbf{z}_1 + \mathbf{z}_2)^T \bar{\mathbf{z}}_3 \\ &= (\mathbf{z}_1^T + \mathbf{z}_2^T)\bar{\mathbf{z}}_3 \\ &= \mathbf{z}_1^T \bar{\mathbf{z}}_3 + \mathbf{z}_2^T \bar{\mathbf{z}}_3 \\ &= \langle \mathbf{z}_1, \mathbf{z}_3 \rangle + \langle \mathbf{z}_2, \mathbf{z}_3 \rangle. \end{aligned}$$

   The proof is similar when addition is in the second component. (But not identical -- you'll need the fact that the complex conjugate is distributive, rather than the transpose.)

2. *These again follow from writing the inner product as a matrix product.*

$$\langle \alpha\mathbf{z}_1, \mathbf{z}_2 \rangle = (\alpha\mathbf{z}_1)^T \bar{\mathbf{z}}_2 = \alpha(\mathbf{z}_1^T \bar{\mathbf{z}}_2) = \alpha\langle \mathbf{z}_1, \mathbf{z}_2 \rangle,$$

   and

$$\langle \mathbf{z}_1, \alpha\mathbf{z}_2 \rangle = \mathbf{z}_1^T \overline{\alpha\mathbf{z}_2} = \mathbf{z}_1^T (\bar{\alpha}\bar{\mathbf{z}}_2) = \bar{\alpha}(\mathbf{z}_1^T \bar{\mathbf{z}}_2) = \alpha\langle \mathbf{z}_1, \mathbf{z}_2 \rangle.$$

3. Note that for any vectors $\mathbf{z}, \mathbf{w}$, $\mathbf{z}^T \mathbf{w}$ is a number, and therefore equal to its own transpose. Thus, we have $\mathbf{z}^T \mathbf{w} = (\mathbf{z}^T \mathbf{w})^T = \mathbf{w}^T \mathbf{z}$, and

$$\overline{\langle \mathbf{z}_1, \mathbf{z}_2 \rangle} = \overline{\mathbf{z}_1^T \bar{\mathbf{z}}_2} = \overline{\bar{\mathbf{z}}_2^T \mathbf{z}_1} = \mathbf{z}_2^T \overline{\mathbf{z}_1} = \langle \mathbf{z}_2, \mathbf{z}_1 \rangle.$$

4. This was already demonstrated above.

■

Definition 4.4.4

The **norm** of a vector $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ in ^{n} is given by

$$\|\mathbf{z}\| = \sqrt{\langle \mathbf{z}, \mathbf{z} \rangle} = \sqrt{|z_1|^2 + |z_2|^2 + \cdots + |z_n|^2}.$$

Note that much like the real norm, the complex norm satisfies $\|\alpha \mathbf{z}\| = |\alpha| \|\mathbf{z}\|$ for any (complex) scalar α.

4.4.3 Complex matrices

Linear transformations are defined in exactly the same way, and a complex matrix is simply a matrix whose entries are complex numbers. There are two important operations defined on complex matrices: the conjugate, and the conjugate transpose (also known as the hermitian transpose).

Definition 4.4.5

The **conjugate** of a matrix $A = [a_{ij}] \in M_{mn}()$ is the matrix $\bar{A} = [\bar{a}_{ij}]$. The **conjugate transpose** of A is the matrix A^H defined by

$$A^H = (\bar{A})^T = \overline{(A^T)}.$$

Note that many textbooks use the notation $A^\dagger$ for the conjugate transpose.

Definition 4.4.6

An $n \times n$ matrix $A \in M_{nn}()$ is called **hermitian** if $A^H = A$, and **unitary** if $A^H = A^{-1}$. (A matrix is **skew-hermitian** if $A^H = -A$.)

Hermitian and unitary matrices (or more accurately, linear operators) are very important in quantum mechanics. Indeed, hermitian matrices represent "observable" quantities, in part because their eigenvalues are real, as we'll soon see. For us, hermitian and unitary matrices can simply be viewed as the complex counterparts of symmetric and orthogonal matrices, respectively. In fact, a real symmetric matrix is hermitian, since the conjugate has no effect on it, and similarly, a real orthogonal matrix is technically unitary. As with orthogonal matrices, a unitary matrix can also be characterized by the property that its rows and columns both form orthonormal bases.

Exercise 4.4.7

Show that the matrix $A = \begin{bmatrix} 4 & 1-i & -2+3i \\ 1+i & 5 & 7i \\ -2-3i & -7i & -4 \end{bmatrix}$ is hermitian, and

that the matrix $B = \frac{1}{2}\begin{bmatrix} 1+i & \sqrt{2} \\ 1-i & \sqrt{2i} \end{bmatrix}$ is unitary.

When using SymPy, the hermitian conjugate of a matrix A is executed using A.H. (There appears to also be an equivalent operation named Dagger coming from sympy.physics.quantum, but I've had more success with .H.) The complex unit is entered as I. So for the exercise above, we can do the following.

```
A = Matrix(3,3,[4,1-I,-2+3*I,1+I,5,7*I,-2-3*I,-7*I,-4])
A == A.H
```

```
True
```

The last line verifies that $A = A^H$. We could also replace it with A,A.H to explicitly see the two matrices side by side. Now, let's confirm that B is unitary.

```
B = Matrix(2,2,[1/2+1/2*I,
    sqrt(2)/2,1/2-1/2*I,(sqrt(2)/2)*I])
B,B*B.H
```

```
\[\left(\bbm \frac12+\frac{i}{2}\amp \frac{\sqrt{2}}{2}\\
    \frac12-\frac{i}{2}\amp \frac{\sqrt{2}i}{2}\ebm,
 \bbm \frac12 +
     \left(\frac12-\frac{i}{2}\right)\left(\frac12+\frac{i}{2}\right)\amp
-\frac{i}{2}+\left(\frac12+\frac{i}{2}\right)^2\\
    \left(\frac12-\frac{i}{2}\right)^2\amp
\frac12 +
    \left(\frac12-\frac{i}{2}\right)\left(\frac12+\frac{i}{2}\right)\ebm\right)\]
```

Hmm... That doesn't look like the identity on the right. Maybe try replacing B*B.H with simplify(B*B.H). (You will want to add from sympy import simplify at the top of the cell.) Or you could try B.H, B**-1 to compare results. Actually, what's interesting is that in a Sage cell, B.H == B**-1 yields False, but B.H == simplify(B**-1) yields True!

As mentioned above, hermitian matrices are the complex analogue of symmetric matrices. Recall that a key property of a symmetric matrix is its symmetry with respect to the dot product. For a symmetric matrix A, we had $\mathbf{x} \cdot (A\mathbf{y}) = (A\mathbf{x}) \cdot \mathbf{y}$. Hermtian matrices exhibit the same behaviour with respect to the complex inner product.

> **Theorem 4.4.8**
>
> **An $n \times n$ complex matrix A is Hermitian if and only if**
>
> $$\langle A\mathbf{z}, \mathbf{w} \rangle = \langle \mathbf{z}, A\mathbf{w} \rangle$$
>
> **for any $\mathbf{z}, \mathbf{w} \in^n$**

Proof. Note that the property $A^H = A$ is equivalent to $A^T = \bar{A}$. This gives us

$$\langle A\mathbf{z}, \mathbf{w} \rangle = (A\mathbf{z})^T \bar{\mathbf{w}} = (\mathbf{z}^T A^T)\bar{\mathbf{w}} = (\mathbf{z}^T \bar{A})\bar{\mathbf{w}} = \mathbf{z}^T (\overline{A\mathbf{w}}) = \langle \mathbf{z}, \mathbf{w} \rangle.$$

Conversely, suppose $\langle A\mathbf{z}, \mathbf{w} \rangle = \langle \mathbf{z}, A\mathbf{w} \rangle$ for all $\mathbf{z}, \mathbf{w} \in^n$, and let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ denote the standard basis for ^{n}. Then

$$a_{ji} = \langle A\mathbf{e}_i, \mathbf{e}_j \rangle = \langle \mathbf{e}_i, A\mathbf{e}_j \rangle = \overline{a_{ij}},$$

which shows that $A^T = \bar{A}$.   ∎

Next, we've noted that one advantage of doing linear algebra over is that every polynomial can be completely factored, including the characteristic polynomial. This means that we can always find eigenvalues for a matrix. When that matrix is Hermitian, we get a surprising result.

> **Theorem 4.4.9**
>
> **For any hermitian matrix A,**
>
> 1. **The eigenvalues of A are real.**
>
> 2. **Eigenvectors corresponding to distinct eigenvalues are orthogonal.**

Proof.

1. *Suppose $A\mathbf{z} = \lambda\mathbf{z}$ for some $\lambda \in$ and $\mathbf{z} \neq \mathbf{0}$. Then*

$$\lambda\langle\mathbf{z}, \mathbf{z}\rangle = \langle\lambda\mathbf{z}, \mathbf{z}\rangle = \langle A\mathbf{z}, \mathbf{z}\rangle = \langle\mathbf{z}, A\mathbf{z}\rangle = \langle\mathbf{z}, \lambda\mathbf{z}\rangle = \bar{\lambda}\langle\mathbf{z}, \mathbf{z}\rangle.$$

   Thus, $(\lambda - \bar{\lambda})\|z\|^2 = 0$, and since $\|z\| \neq 0$, we must have $\bar{\lambda} = \lambda$, which means $\lambda \in \mathbb{R}$.

2. *Similarly, suppose λ_1, λ_2 are eigenvalues of A, with corresponding eigenvectors $\mathbf{z}, \mathbf{w}$. Then*

$$\lambda_1\langle\mathbf{z}, \mathbf{w}\rangle = \langle\lambda_1\mathbf{z}, \mathbf{w}\rangle = \langle A\mathbf{z}, \mathbf{w}\rangle = \langle\mathbf{z}, A\mathbf{w}\rangle = \langle\mathbf{z}, \lambda_2\mathbf{w}\rangle = \bar{\lambda}_2\langle\mathbf{z}, \mathbf{w}\rangle.$$

   This gives us $(\lambda_1 - \bar{\lambda}_2)\langle\mathbf{z}, \mathbf{w}\rangle = 0$. And since we already know λ_2 must be real, and $\lambda_1 \neq \lambda_2$, we must have $\langle\mathbf{z}, \mathbf{w}\rangle = 0$.

∎

In light of Theorem ??, we realize that diagonalization of hermitian matrices will follow the same script as for symmetric matrices. Indeed, Gram-Schmidt Orthonormalization Algorithm applies equally well in ^{n}, as long as we replace the dot product with the complex inner product. This suggests the following.

> **Theorem 4.4.10 Spectral Theorem.**
>
> **If A is an $n \times n$ hermitian matrix, then there exists an orthonormal basis of ^{n} consisting of eigenvectors of A. Moreover, the matrix U whose columns consist of those eigenvectors is unitary, and the matrix $U^H AU$ is diagonal.**

> **Exercise 4.4.11**
>
> *Confirm that the matrix $A = \begin{bmatrix} 4 & 3-i \\ 3+i & 1 \end{bmatrix}$ is hermitian. Then, find the eigenvalues of A, and a unitary matrix U such that $U^H AU$ is diagonal.*

To do the above exercise using SymPy, we first define A and ask for the eigenvectors.

```
A = Matrix(2,2,[4,3-I,3+I,1])
A.eigenvects()
```

```
\[\left[\left(-1,1,\bbm
    -\frac35+\frac{i}{5}\\1\ebm\right),\left(6,1,\bbm
    \frac32-\frac{i}{2}\ebm\right)\right]\]
```

We can now manually determine the matrix U, as we did above, and input it:

```
U = Matrix([[(3-I)/sqrt(35),(3-I)/sqrt(14)],
            [-5/sqrt(35),2/sqrt(14)]])
```

To confirm it's unitary, add the line `U*U.H` *to the above, and confirm that you get the identity matrix as output. You might need to use* `simplify(U*U.H)` *if the result is not clear. Now, to confirm that* $U^H A U$ *really is diagonal, go back to the cell above, and enter it. Try* `(U.H)*A*U`, *just to remind yourself that adding the* `simplify` *command is often a good idea.*

If you want to cut down on the manual labour involved, we can make use of some of the other tools SymPy provides. In the next cell, we're going to assign the output of `A.eigenvects()` *to a list. The only trouble is that the output of the eigenvector command is a list of lists. Each list item is a list* (eigenvalue, multiplicity, [eigenvectors])*.*

```
L = A.eigenvects()
L
```

```
\[\left[\left(-1,1,\bbm -\frac35+\frac{i}{5}\\1\ebm\right),
    \left(6,1,\bbm \frac32-\frac{i}{2}\\1\ebm\right)\right]\]
```

Try the above modifications, in sequence. First, replacing the second line by `L[0]` *will give the first list item, which is another list:*

$$\left(-1, 1, \left[\left[-\tfrac{3}{5} + \tfrac{i}{5}\right]\right]\right).$$

We want the third item in the list, so try `(L[0])[2]`*. But note the extra set of brackets! There could (in theory) be more than one eigenvector, so this is a list with one item. To finally get the vector out, try* `((L[0])[2])[0]`*. (There is probably a better way to do this. Someone who is more fluent in Python is welcome to advise.)*

Now that we know how to extract the eigenvectors, we can normalize them, and join them to make a matrix. The norm of a vector is simnply `v.norm()`*, and to join column vectors* `u1` *and* `u2` *to make a matrix, we can use the command* `u1.row_join(u2)`*. We already defined the matrix* `A` *and list* `L` *above, but here is the whole routine in one cell, in case you didn't run all the cells above.*

```
from sympy import *
init_printing()
A = Matrix(2,2,[4,3-I,3+I,1])
L = A.eigenvects()
v = ((L[0])[2])[0]
w = ((L[1])[2])[0]
u1 = (1/v.norm())*v
u2 = (1/w.norm())*w
U = u1.row_join(u2)
u1, u2, U, simplify(U.H*A*U)
```

```
\[\left(\bbm\frac{\sqrt{35}(-\frac35+\frac{i}{5})}{7}\\
    \frac{\sqrt{35}}{7}\ebm,\bbm
    \frac{\sqrt{14}(\frac32-\frac{i}{2})}{7}\\
```

```
    \frac{\sqrt{14}}{7}\ebm,
 \bbm \frac{\sqrt{35}(-\frac35+\frac{i}{5})}{7} \amp
    \frac{\sqrt{14}(\frac32 -\frac{i}{2})}{7}\\
    \frac{\sqrt{35}}{7} \amp\frac{\sqrt{14}}{7}\ebm,
 \bbm -1\amp 0\\0\amp 6\ebm\right)\]
```

Believe me, you want the simplify command on that last matrix.

While Theorem ?? guarantees that any hermitian matrix can be "unitarily diagonalized", there are also non-hermitian matrices for which this can be done as well. A classic example of this is given in Nicholson's book, so we do not repeat the details here: the matrix $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ is a real matrix with complex eigenvalues $\pm i$, and while it is neither symmetric nor hermitian, it can be orthogonally diagonalized. This should be contrasted with the real spectral theorem, where any matrix that can be orthogonally diagonalized is necessarily symmetric.

This suggests that perhaps hermitian matrices are not quite the correct class of matrix for which the spectral theorem should be stated. Indeed, it turns out there is a somewhat more general class of matrix: the normal matrices.

Definition 4.4.12

*An $n \times n$ matrix A is **normal** if $A^H A = AA^H$.*

It turns out that a matrix A is normal if and only if $A = UDU^H$ for some unitary matrix U and diagonal matrix D. A further generalization is known as Schur's Theorem.

Theorem 4.4.13

For any complex $n \times n$ matrix A, there exists a unitary matrix U such that $U^H AU = T$ is upper-triangular, and such that the diagonal entries of T are the eigenvalues of A.

Using Schur's Theorem, we can obtain a famous result, known as the Cayley-Hamilton Theorem, for the case of complex matrices. (It is true for real matrices as well, but we don't yet have the tools to prove it.) The Cayley-Hamilton Theorem states that substituting any matrix into its characteristic polynomial results in the zero matrix. To understand this result, we should first explain how to define a polynomial of a matrix.

Given a polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x_n$, we define $p(A)$ as

$$p(A) = a_0 I + a_1 A + \cdots + a_n A^n.$$

(Note the presence of the identity matrix in the first term, since it does not make sense to add a scalar to a matrix.) Note further that since $(P^{-1}AP)^n = P^{-1}A^n P$ for any invertible matrix P and positive integer n, we have $p(U^H AU) = U^H p(A)U$ for any polynomial p and unitary matrix U.

Theorem 4.4.14

Let A be an $n \times n$ complex matrix, and let $c_A(x)$ denote the characteristic polynomial of A. Then we have $c_A(A) = 0$.

Proof. *By Theorem ??, there exists a unitary matrix U such that $A = UTU^H$, where T is upper triangular, and has the eigenvalues of A as diagonal entries. Since $c_A(A) = c_A(UTU^H) = Uc_A(T)U^H$, and $c_A(x) = c_T(x)$ (since A and T are similar) it suffices to show that $c_A(A) = 0$ when A is upper-triangular. (If you like, we are showing that $C_T(T) = 0$, and deducing that $c_A(A) = 0$.) But if A is upper-triangular, so is xI_A,*

and therefore, $\det(xI - A)$ is just the product of the diagonal entries. That is,

$$c_A(x) = (x - \lambda_1)(x - \lambda_2) \cdots (x - \lambda_n),$$

so

$$c_A(A) = (A - \lambda_1 I)(A - \lambda_2 I) \cdots (A - \lambda_n I).$$

Since the first column of A is $\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \end{bmatrix}^T$, the first column of $A - \lambda_1 I$ is identically zero. The second column of $A - \lambda_2 I$ similarly has the form $\begin{bmatrix} k & 0 & \cdots & 0 \end{bmatrix}$ for some number k.

It follows that the first two columns of $(A - \lambda_1 I)(A - \lambda_2 I)$ are identically zero. Since only the first two entries in the third column of $(A - \lambda_3 I)$ can be nonzero, we find that the first three columns of $(A - \lambda_1 I)(A - \lambda_2 I)(A - \lambda_3 I)$ are zero, and so on. ∎

4.5 Matrix Factorizations and Eigenvalues

This section is a rather rapid tour of some cool ideas that get a lot of use in applied linear algebra. We are rather light on details here. The interested reader can consult sections 8.3–8.6 in Nicholson.

4.5.1 Matrix Factorizations

*Recall that an $n \times n$ matrix A is symmetric if $A^T = A$ and hermitian if $A^H = A$, where A^H is the conjugate transpose of a complex matrix. In either case, the corresponding matrix transformation T_A is said to be **self-adjoint**, which means that it satisfies the condition*

$$\langle u, T_A v \rangle = \langle T_A u, v \rangle$$

for all $u, v \in \mathbb{R}^n$ (or $\mathbb{C}^n$).

All such matrices (or operators) can be diagonalized, in the sense that there is an orthonormal basis of eigenvectors for that matrix. These eigenvectors can be arranged to form an orthogonal matrix P (or unitary matrix U) such that

$$P^T A P = D \quad \text{(or } U^H A U = D\text{)},$$

where D is a diagonal matrix whose entries are the eigenvalues of A.

4.5.1.1 Positive Operators

> **Definition 4.5.1**
>
> *A symmetric/hermitian matrix A (or operator T) is **positive** if $\mathbf{x}^T A \mathbf{x} \geq 0$ ($\langle \mathbf{x}, T\mathbf{x} \rangle \geq 0$) for all vectors $\mathbf{x} \neq \mathbf{0}$. It is **positive-definite** if $\mathbf{x}^T A \mathbf{x} > 0$ for all nonzero $\mathbf{x}$.*

This is equivalent to requiring that all the eigenvalues of A are non-negative. Every positive matrix A has a unique positive square root: a matrix R such that $R^2 = A$.

Some books will define positive-definite operators by the condition $\mathbf{x}^T A\mathbf{x}$ without the requirement that A must be symmetric/hermitian. However, we will stick to the simpler definition.

> **Theorem 4.5.2**
>
> **For any $n \times n$ matrix U, the matrix $A = U^T U$ is positive. Moreover, if U is invertible, then A is positive-definite.**

Proof. *For any* $\boldsymbol{x} \neq \boldsymbol{0}$ *in* $\mathbb{R}^n$,

$$\boldsymbol{x}^T A \boldsymbol{x} = \boldsymbol{x}^T U^T U \boldsymbol{x} = (U\boldsymbol{x})^T (U\boldsymbol{x}) = \|U\boldsymbol{x}\|^2 \geq 0.$$

∎

What is interesting is that the converse to the above statement is also true. The **Cholesky factorization** *of a positive-definite matrix* A *is given by* $A = U^T U$, *where* U *is upper-triangular, with positive diagonal entries. (See Nicholson for details.)*

Even better is that there is a very simple algorithm for obtaining the factorization: Carry the matrix A *to triangular form, using only row operations of the type* $R_i + kR_j \to R_i$. *Then divide each row by the square root of the diagonal entry.*

The SymPy library contains the `cholesky()` *algorithm. Note however that it produces a lower triangular matrix, rather than upper triangular. (That is, the output gives* $L = U^T$ *rather than* U, *so you will have* $A = LL^T$.) *Let's give it a try. First, enter a positive-definite matrix. (We'll try the one from Example 8.3.3 in Nicholson.)*

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix([[10,5,2],[5,3,2],[2,2,3]])
A
```

\[\bbm 10\amp 5\amp 2\\5\amp 3\amp 2\\2\amp 2\amp 3\ebm\]

Next, find the Cholesky factorization:

```
L = A.cholesky()
L, L*L.T
```

\[\left(\bbm \sqrt{10}\amp 0\amp 0\\ \frac{\sqrt{10}}{2}
    \amp \frac{\sqrt{2}}{2} \amp 0\\\frac{\sqrt{10}}{5} \amp
    \sqrt{2}\amp \frac{\sqrt{15}}{5}\ebm, \bbm 10\amp 5\amp
    2\\ 5\amp 3\amp 2\\ 2\amp 2\amp 3\ebm\right)\]

```
L*L.T == A
```

```
True
```

4.5.1.2 Singular Value Decomposition

For any $n \times n$ *matrix* A, *the matrices* $A^T A$ *and* AA^T *are both positive. (Exercise!) This means that we can define* $\sqrt{A^T A}$, *even if* A *itself is not symmetric or positive.*

- *Since* $A^T A$ *is symmetric, we know that it can be diagonalized.*

- *Since* $A^T A$ *is positive, we know its eigenvalues are non-negative.*

- *This means we can define the* **singular values** $\sigma_i = \sqrt{\lambda_i}$ *for each* $i = 1, \ldots, n$.

- **Note:** *it's possible to do this even if* A *is not a square matrix!*

The SymPy library has a function for computing the singular values of a matrix. Given a matrix `A`, *the command* `A.singular_values()` *will return its singular values. Try this for a few different matrices below:*

```
A = Matrix([[1,2,3],[4,5,6]])
A.singular_values()
```

\[\left[\sqrt{\frac{\sqrt{8065}}{2}+\frac{91}{2}},\sqrt{\frac{91}{2}-\frac{\sqrt{8065}}{2}},0\right]

In fact, SymPy can even return singular values for a matrix with variable entries!
Try the following example from the SymPy documentation[1].

```
from sympy import Symbol
x = Symbol('x', real=True)
M = Matrix([[0,1,0],[0,x,0],[-1,0,0]])
M,M.singular_values()
```

\[\left(\bbm 0\amp 1\amp 0\\ 0\amp x\amp 0\\ -1\amp 0\amp
    0\ebm , \left[\sqrt{x^2+1}, 1, 0\right]\right)\]

For an $n \times n$ matrix A, we might not be able to diagonalize A (with a single ortho-normal basis). However, it turns out that it's always possible to find a pair of ortho-normal bases $\{e_1, \ldots, e_n\}, \{f_1, \ldots, f_n\}$ such that

$$Ax = \sigma_1(x \cdot e_1)f_1 + \cdots + \sigma_n(x \cdot e_n)f_n.$$

In matrix form, $A = P\Sigma_A Q^T$ for orthogonal matrices P, Q.

In fact, this can be done even if A is not square, which is arguably the more inter-esting case! Let A be an $m \times n$ matrix. We will find an $m \times m$ orthogonal matrix P and $n \times n$ orthogonal matrix Q, such that $A = P\Sigma_A Q^T$, where Σ_A is also $m \times n$.

The basis $\{f_1, \ldots, f_n\}$ is an orthonormal basis for $A^T A$, and the matrix Q is the matrix whose columns are the vectors f_i. As a result, Q is orthogonal.

The matrix Σ_A is the same size as A. First, we list the positive singular values of A in decreasing order:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0.$$

Then, we let $D_A = diag(\sigma_1, \ldots, \sigma_k)$, and set

$$\Sigma_A = \begin{bmatrix} D_A & 0 \\ 0 & 0 \end{bmatrix}.$$

That is, we put D_A in the upper-left, and then fill in zeros as needed, until Σ_A is the same size as A.

Next, we compute the vectors $e_i = \frac{1}{\|Af_i\|}Af_i$, for $i = 1, \ldots, k$. As shown in Nicolson, $\{e_1, \ldots, e_r\}$ will be an orthonormal basis for the column space of A. The matrix P is constructed by extending this to an orthonormal basis of $\mathbb{R}^m$.

All of this is a lot of work to do by hand, but it turns out that it can be done numer-ically, and more importantly, efficiently, by a computer. The SymPy library does not have an svD algorithm, but the mpmath library does, and it works well with SymPy.

Example 8.6.1 in Nicolson shows that for the matrix $A = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$ we should have

$$P = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \Sigma_A = \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, Q = \frac{1}{\sqrt{6}} \begin{bmatrix} 2 & -1 & 1 \\ 0 & \sqrt{3} & \sqrt{3} \\ -\sqrt{2} & -\sqrt{2} & \sqrt{2} \end{bmatrix}.$$

Let us test this on the computer. First, we import the mpmath library, and define A. (Importantly, the svd routine from mpmath will accept a SymPy matrix as input. We

If A is symmetric and positive-definite, the singular values of A are just the eigenvalues of A, and the singular value de-composition is the same as di-agonalization.

---

[1]docs.sympy.org/latest/modules/matrices/matrices.html#sympy.matrices.
matrices.MatrixEigen.singular_values

do not, for example, have to first convert it to a NumPy array.) Since both SymPy and mpmath have similar functions, we `import` *as for both libraries, so we can distinguish between them as needed.*

```
import sympy as sy
import mpmath as mp
sy.init_printing()
mp.pretty = True
A = mp.matrix([[1,0,1],[-1,1,0]])
A
```

```
matrix(
[['1.0', '0.0', '1.0'],
 ['-1.0', '1.0', '0.0']])
```

Note that we defined A *as a mpmath* `matrix`*. The* `svd` *command from the mpmath library will also work on a SymPy* `Matrix`*, but certain mpmath commands, like the* `chop` *command used to round decimals, will not. Next, we apply the* `svd` *algorithm.*

```
P,S,QT = mp.svd(A)
P,S,QT
```

```
(matrix(
[['-0.707106781186547', '0.707106781186547'],
 ['0.707106781186547', '0.707106781186547']]),
 matrix(
[['1.73205080756888'],
 ['1.0']]),
 matrix(
[['-0.816496580927726', '0.408248290463863',
     '-0.408248290463863'],
 ['3.14018491736755e-16', '0.707106781186547',
     '0.707106781186548']]))
```

Note that the input isn't quite so nice this time: the algorithm is numerical. For some reason, if you define A *as a SymPy matrix, then* P *will be a SymPy matrix, but* S *and* Q *will not. (I have no idea why.) Also, the matrix* S *is a column matrix, containing the positive singular values of* A*. We can turn it into a diagonal matrix as follows, using the* `diag` *command from mpmath:*

```
S1 = mp.diag(S)
S1
```

```
matrix(
[['1.73205080756888', '0.0'],
 ['0.0', '1.0']])
```

In case you don't recognize the square root of 3 by its decimal approximation, we can check:

```
S1*S1
```

```
matrix(
[['1.73205080756888', '0.0'],
 ['0.0', '1.0']])
```

The matrix Σ_A is supposed to be 2×3, with an additional column of zeros. If we want to define this as a SymPy matrix, we can. Note that we are using the `diag` command from SymPy this time, rather than mpmath.

```
S2 = sy.Matrix(S1)
SigA = S2.row_join(sy.Matrix([0,0]))
SigA
```

\[\bbm 1.73205080756888 \amp 0.0 \amp 0\\ 0.0 \amp 1.0 \amp
    0\ebm\]

We will see in a minute that this matrix is not really necessary, because the `svd` algorithm in mpmath is a bit different from the one in Nicholson.

The matrix P is already a SymPy matrix, as it turns out, but Q is not.

```
QT
```

```
matrix(
[['-0.816496580927726', '0.408248290463863',
    '-0.408248290463863'],
 ['3.14018491736755e-16', '0.707106781186547',
    '0.707106781186548']])
```

Notice that this matrix is not square! The output from the algorithm also has the transpose already applied.

```
Q1 = QT.T
Q1
```

```
matrix(
[['-0.816496580927726', '3.14018491736755e-16'],
 ['0.408248290463863', '0.707106781186547'],
 ['-0.408248290463863', '0.707106781186548']])
```

The matrix `Q1` we get here is (up to a difference in sign) the first two columns of the matrix Q found in Nicholson. This makes sense: the matrix S that we get from the algorithm is D_A, not Σ_A. The third column of Σ_A is zero. So when we multiply by Q^T, the third row of Q^T (that is, the third column of Q) is lost. That is, $S(QT) = \Sigma_A Q^T$.

To confirm that everything worked, we can multiply:

```
P*S1*QT
```

```
matrix(
[['0.999999999999999', '-3.32977184235283e-17',
    '0.999999999999999'],
 ['-1.0', '1.0', '1.37483317152544e-16']])
```

Maybe that's not so enlightening, given all the decimal places. Let's check the difference with the matrix A:

```
A-P*S1*QT
```

```
matrix(
[['5.55111512312578e-16', '3.32977184235283e-17',
```

```
    '8.88178419700125e-16'],
  ['-3.33066907387547e-16', '1.11022302462516e-16',
    '-1.37483317152544e-16']])
```

Looks like some pretty small numbers there. The mpmath library includes the chop *command for truncating decimals.*

```
mp.chop(A-P*S1*QT)
```

```
 matrix(
 [['0.0', '0.0', '0.0'],
  ['0.0', '0.0', '0.0']])
```

The Singular Value Decomposition has a lot of useful appplications, some of which are described in Nicholson's book. On a very fundamental level the SVD *provides us with information on some of the most essential properties of the matrix A, and any system of equations with A as its coefficient matrix.*

Recall the following definitions for an $m \times n$ matrix A:

1. *The **rank** of A is the number of leadning ones in the* RREF *of A, which is also equal to the dimension of the column space of A (or if you prefer, the dimension of $im(T_A)$).*

2. *The **column space** of A, denoted $col(A)$, is the subspace of $\mathbb{R}^m$ spanned by the columns of A. (This is the image of the matrix transformation T_A; it is also the space of all vectors $\boldsymbol{b}$ for which the system $A\boldsymbol{x} = \boldsymbol{b}$ is consistent.)*

3. *The **row space** of A, denoted $row(A)$, is the span of the rows of A, viewed as column vectors in $\mathbb{R}^n$.*

4. *The **null space** of A is the space of solutions to the homogeneous system $A\boldsymbol{x} = \boldsymbol{0}$. This is, of course, equal the kernel of the associated transformation T_A.*

There are some interesting relationships among these spaces, which are left as an exercise. (Solutions are in Nicholson if you get stumped.)

> **Exercise 4.5.3**
>
> *Let A be an $m \times n$ matrix. Prove the following:*
>
> 1. *$(row(A))^{\perp} = null(A)$*
>
> 2. *$(col(A))^{\perp} = null(A^T)$*

Here's the cool thing about the SVD*. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ be the positive singular values of A. Let $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r, \ldots, \boldsymbol{q}_n$ be the orthonormal basis of eigenvectors for $A^T A$, and let $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_r, \ldots, \boldsymbol{p}_m$ be the orthonormal basis of $\mathbb{R}^m$ constructed in the* SVD *algorithm. Then:*

1. *$rank(A) = r$*

2. *$\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r$ form a basis for $row(A)$.*

3. *$\boldsymbol{p}_1, \ldots, \boldsymbol{p}_r$ form a basis for $col(A)$ (and thus, the "row rank" and "column rank" of A are the same).*

4. *$\boldsymbol{q}_{r+1}, \ldots, \boldsymbol{q}_n$ form a basis for $null(A)$. (And these are therefore the basis solutions of $A\boldsymbol{x} = \boldsymbol{0}$!)*

5. *$\boldsymbol{p}_{r+1}, \ldots, \boldsymbol{p}_m$ form a basis for $null(A^T)$.*

If you want to explore this further, have a look at the excellent notebook by Dr. Juan H Klopper[2]. The `ipynb` file can be found on his GitHub page[3]. In it, he takes you through various approaches to finding the singular value decomposition, using the method above, as well as using NumPy and SciPy (which, for industrial applications, are superior to SymPy and mpmath).

4.5.1.3 Polar Decomposition

For any $n \times n$ matrix A, there exists an orthogonal (or unitary) matrix P such that

$$A = P\sqrt{A^T A}.$$

This is meant to remind you of the polar decomposition

$$z = e^{i\theta}\sqrt{\bar{z}z}$$

for a complex number.

One way to compute the polar decomposition is using the Singular Value Decomposition (see Nicholson's text). Note that both P and $\sqrt{A^T A}$ can be diagonalized, but usually not with the same orthonormal basis.

4.5.1.4 QR Factorization

Suppose A is an $m \times n$ matrix with independent columns. (Question: for this to happen, which is true — $m \geq n$, or $n \geq m$?)

A QR-factorization of A is a factorization of the form $A = QR$, where Q is $m \times n$, with orthonormal columns, and R is an invertible upper-triangular ($n \times n$) matrix with positive diagonal entries. If A is a square matrix, Q will be orthogonal.

A lot of the methods we're looking at here involve more sophisticated numerical techniques than SymPy is designed to handle. If we wanted to spend time on these topics, we'd have to learn a bit about the NumPy package, which has built in tools for finding things like polar decomposition and singular value decomposition. However, SymPy does know how to do QR factorization. After defining a matrix A, we can use the command

```
Q, R = A.QRdecomposition()
```

.

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix(3,3,[1,-2,3,3,-1,2,4,2,5])
Q, R = A.QRdecomposition()
A, Q, R
```

```
\[\left(\bbm 1\amp -2\amp 3\\ 3\amp -1\amp 2\\ 4\amp 2\amp
    5\ebm, \bbm \frac{\sqrt{26}}{6} \amp
    -\frac{11\sqrt{26}}{78} \amp \frac23\\
    \frac{3\sqrt{26}}{26} \amp -\frac{7\sqrt{26}}{78} \amp
    -\frac23\\ \frac{2\sqrt{26}}{13} \amp
    \frac{4\sqrt{26}}{39} \amp \frac13\ebm, \bbm
    \sqrt{26}\amp \frac{3\sqrt{26}}{26} \amp
```

---

[2] `www.juanklopper.com/wp-content/uploads/2015/03/III_05_Singular_value_decomposition.html`

[3] `github.com/juanklopper/MIT_OCW_Linear_Algebra_18_06`

```
\frac{29\sqrt{26}}{26}\\ 0\amp \frac{15\sqrt{26}}{26}
\amp -\frac{7\sqrt{26}}{78}\\0\amp 0\amp
\frac73\ebm\right)\]
```

Let's check that the matrix Q really is orthogonal:

```
Q**(-1) == Q.T
```

```
True
```

Details of how to perform the QR factorization can be found in Nicholson's text-book. It's essentially a consequence of performing the Gram-Schmidt algorithm on the columns of A, and keeping track of our work.

The calculation above is a symbolic computation, which is nice for understanding what's going on. The reason why the QR factorization is useful in practice is that there are efficient numerical methods for doing it (with good control over rounding errors). Our next topic looks at a useful application of the QR factorization.

4.5.2 Computing Eigenvalues

Our first method focuses on the dominant eigenvalue of a matrix. An eigenvalue is dominant if it is larger in absolute value than all other eigenvalues. For example, if A has eigenvalues $1, 3, -2, -5$, then -5 is the dominant eigenvalue.

If A has eigenvalues $1, 3, 0, -4, 4$ then there is no dominant eigenvalue. Any eigenvector corresponding to a dominant eigenvalue is called a dominant eigenvector.

4.5.2.1   The Power Method

If a matrix A has a dominant eigenvalue, there is a method for finding it (approximately) that does not involve finding and factoring the characteristic polynomial of A.

We start with some initial guess x_0 for a dominant eigenvector. We then set $x_{k+1} = Ax_k$ for each $k \geq 0$, giving a sequence

$$x_0, Ax_0, A^2x_0, A^3x_0, \ldots.$$

We expect (for reasons we'll explain) that $\|x_k - x\| \to 0$ as $k \to \infty$, where x is a dominant eigenvector. Let's try an example.

```
A = Matrix(2,2,[1,-4,-3,5])
A,A.eigenvects()
```

```
\[\left(\bbm 1\amp -4\\-3\amp 5\ebm, \left[\left(-1, 1,
    \left[\bbm 2\\1\ebm\right]\right),\left(7, 1, \left[\bbm
    -\frac23\\1\ebm\right]\right)\right]\right)\]
```

The dominant eigenvalue is $\lambda = 7$. Let's try an initial guess of $x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and see what happens.

```
x0 = Matrix(2,1,[1,0])
L = list()
for k in range(10):
    L.append(A**k*x0)
L
```

```
\begin{align*}
\left[\bbm 1\\0\ebm, \bbm 1\\-3\ebm, \bbm
    13\\-18\ebm,\right. \amp \bbm 85\\-129\ebm, \bbm
    601\\-900\ebm, \bbm 4201\\-6303\ebm,\\ \amp \left.\bbm
    29413\\ -44118\ebm, \bbm 205885\\-308829\ebm, \bbm
    1441201\\-2161800\ebm, \bbm
    10088401\\-15132603\ebm\right]
\end{align*}
```

We might want to confirm whether that rather large fraction is close to $\frac{2}{3}$. To do so, we can get the computer to divide the numerator by the denominator.

```
L[9][0]/L[9][1]
```

```
\[-\frac{10088401}{15132603}, \text{ or }
    -0.666666600584182\]
```

The above might show you the fraction rather than its decimal approximation. (This may depend on whether you're on Sage or Jupyter.) To get the decimal, try wrapping the above in `float()` (or `N`, or append with `.evalf()`).

For the eigenvalue, we note that if $Ax = \lambda x$, then

$$\frac{x \cdot Ax}{\|x\|^2} = \frac{x \cdot (\lambda x)}{\|x\|^2} = \lambda.$$

This leads us to consider the Rayleigh quotients

$$r_k = \frac{x_k \cdot x_{k+1}}{\|x_k\|^2}.$$

```
M = list()
for k in range(9):
    M.append((L[k].dot(L[k+1]))/(L[k].dot(L[k])))
M
```

```
\begin{align*}
\left[1,\frac{67}{10}, \frac{3427}{493},
    \frac{167185}{23866}\right.,\amp\frac{8197501}{1171201},\frac{401639767}{57376210},\\
    \amp
    \left.\frac{19680613327}{2811522493},\frac{964348200085}{137763984466},\frac{47253074775001}{675
\end{align*}
```

We can convert a rational number r to a float using either `N(r)` or `r.evalf()`. (The latter seems to be the better bet when working with a list.)

```
M2 = list()
for k in range(9):
    M2.append((M[k]).evalf())
M2
```

```
\begin{align*}1.0,\right.\amp 6.7,6.95131845841785,\\ \amp
    7.00515377524512, 6.99922643508672, 7.00010974931945,\\
    \amp \qquad \left. 6.9999843060121, 7.00000224168168,
    6.9999996797533\right]\end{align*}
```

4.5.2.2 The QR Algorithm

Given an $n \times n$ matrix A, we know we can write $A = QR$, with Q orthogonal and R upper-triangular. The QR-algorithm exploits this fact. We set $A_1 = A$, and write $A_1 = Q_1 R_1$.

Then we set $A_2 = R_1 Q_1$, and factor: $A_2 = Q_2 R_2$. Notice $A_2 = R_1 Q_1 = Q_1^T A_1 Q_1$. Since A_2 is similar to A_1, A_2 has the same eigenvalues as $A_1 = A$.

Next, set $A_3 = R_2 Q_2$, and factor as $A_3 = Q_3 R_3$. Since $A_3 = Q_2^T A_2 Q_2$, A_3 has the same eigenvalues as A_2. In fact, $A_3 = Q_2^T (Q_1^T A Q_1) Q_2 = (Q_1 Q_2)^T A (Q_1 Q_2)$.

After k steps we have $A_{k+1} = (Q_1 \cdots Q_k)^T A (Q_1 \cdots Q_k)$, which still has the same eigenvalues as A. By some sort of dark magic, this sequence of matrices converges to an upper triangular matrix with eigenvalues on the diagonal!

Consider the matrix $A = \begin{bmatrix} 5 & -2 & 3 \\ 0 & 4 & 0 \\ 0 & -1 & 3 \end{bmatrix}$

```
A = Matrix(3,3,[5,-2,3,0,4,0,0,-1,3])
A.eigenvals()
```

\[\{3:1, 4:1, 5:1\}\]

```
Q1,R1 = A.QRdecomposition()
A2=R1*Q1
A2,Q1,R1
```

\[\left(\bbm 5\amp -\frac{11\sqrt{17}}{17}\amp
    \frac{10\sqrt{17}}{17}\\ 0\amp \frac{71}{17} \amp
    \frac{5}{17}\\ 0\amp -\frac{12}{17} \amp
    \frac{48}{17}\ebm, \bbm 1\amp 0\amp 0\\ 0\amp
    \frac{4\sqrt{17}}{17} \amp \frac{\sqrt{17}}{17}\\ 0\amp
    -\frac{\sqrt{17}}{17}, \frac{4\sqrt{17}}{17}\ebm, \bbm
    5\amp -2\amp 3\\ 0\amp \sqrt{17}\amp
    -\frac{3\sqrt{17}}{17}\\ 0\amp 0\amp
    \frac{12\sqrt{17}}{17}\ebm\right)\]

Now we repeat the process:

```
Q2,R2 = A2.QRdecomposition()
A3=R2*Q2
A3.evalf()
```

\[\bbm 5.0\amp -3.0347711718635 \amp 1.94683433666715\\
    0\amp 4.20655737704918 \amp 0.527868852459016\\ 0 \amp
    -0.472131147540984 \amp 2.79344262295082\ebm\]

Do this a few more times, and see what results! (If someone can come up with a way to code this as a loop, let me know!) The diagonal entries should get closer to $5, 4, 3$, respectively, and the $(3, 2)$ entry should get closer to 0.

4.6 Worksheet: Singular Value Decomposition

For this worksheet, the reader is directed to Section 8.6 of Linear Algebra with Applications, by Keith Nicholson, and, of course, to Section ??. (See also notebook by Dr. Juan H Klopper[1].)

In Section ?? we saw that the svd algorithm in the mpmath library does things a little bit differently than Nicholson. If we start with a square matrix A, the results are the same, but if A is not square, the decomposition $A = P\Sigma_A Q^T$ looks a little different. In particular, if A is $m \times n$, the matrix Σ_A defined in Nicholson will also be $m \times n$, but it will contain some rows or columns of zeros that are added to get the desired size. The matrix Q is an orthogonal $n \times n$ matrix whose columns are an orthonormal basis of eigenvectors for $A^T A$. The matrix P is an orthogonal $m \times m$ matrix whose columns are an orthonormal basis of $\mathbb{R}^m$. (The first r columns of P are given by $A\boldsymbol{q}_i$, where $\boldsymbol{q}_i$ is the eigenvector of $A^T A$ corresponding to the positive singular value σ_i.)

The svd algorithm provided by mpmath replaces Σ_A by the $m \times m$ diagonal matrix of singular values. The matrix Q is replaced by the $m \times n$ matrix whose columns are the first m eigenvectors of $A^T A$. (Note that the rank of $A^T A$ is equal to the rank of A, which is equal to the number of nonzero eigenvectors of $A^T A$ (counted with multiplicity).) So we will have $m \geq r$, where r is the number of nonzero singular values.

The product $\Sigma_A Q^T$ will be the same in both cases, and the matrix P is the same as well. This time, rather than using the mpmath algorithm, we will work through the process as outlined in Nicholson step-by-step. First, we will work through (again) Example 8.6.1 in Nicholson. Let $A = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$. First, we get the singular values:

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix([[1,0,1],[-1,1,0]])
L0=A.singular_values()
L0
```

Next, we get the eigenvalues and eigenvectors of $A^T A$:

```
B = (A.T)*A
L1=B.eigenvects()
L1
```

Now we need to normalize the eigenvectors, in the correct order. Note that the eigenvectors were listed in increasing order of eigenvalue, so we need to reverse the order. Note that L1 is a list of lists. The eigenvector is the third entry (index 2) in the list (eigenvalue, multiplicity, eigenvector). We also need to turn list elements into matrices. So, for example the second eigenvector is Matrix(L1[1][2]).

```
R1=Matrix(L1[2][2])
R2=Matrix(L1[1][2])
R3=Matrix(L1[0][2])
Q1 = (1/R1.norm())*R1
Q2 = (1/R2.norm())*R2
Q3 = (1/R3.norm())*R3
Q1,Q2,Q3
```

Next, we can assemble these vectors into a matrix, and confirm that it's orthogonal.

```
from sympy import BlockMatrix
Q = Matrix(BlockMatrix([Q1,Q2,Q3]))
Q,Q*Q.T
```

We've made the matrix Q! Next, we construct Σ_A. This we will do by hand.

```
SigA = Matrix([[L0[0],0,0],[0,L0[1],0]])
SigA
```

Alternatively, you could do `SigA = diag(L0[0],L0[1]).row_join(Matrix([0,0]))`. *Finally, we need to make the matrix P. First, we find the vectors $A\boldsymbol{q}_1, A\boldsymbol{q}_2$ and normalize. (Note that $A\boldsymbol{q}_3 = \boldsymbol{0}$, so this vector is unneeded, as expected.)*

```
S1 = A*Q1
S2 = A*Q2
P1 = (1/S1.norm())*S1
P2 = (1/S2.norm())*S2
P = Matrix(BlockMatrix([P1,P2]))
P
```

Note that the matrix P is already the correct size, because rank$(A) = 2\dim(\mathbb{R}^2)$. In general, for an $m \times n$ matrix A, if rank$(A) = r < m$, we would have to extend the set $\{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_r\}$ to a basis for $\mathbb{R}^m$. Finally, we check that our matrices work as advertised.

```
P*SigA*(Q.T)
```

For convenience, here is all of the above code, with all print commands (except the last one) removed. This can be run as a single code cell.

```
from sympy import Matrix,BlockMatrix,init_printing
init_printing()
A = Matrix([[1,0,1],[-1,1,0]])
B=(A.T)*A
L0=A.singular_values()
L1=B.eigenvects()
R1=Matrix(L1[2][2])
R2=Matrix(L1[1][2])
R3=Matrix(L1[0][2])
Q1 = (1/R1.norm())*R1
Q2 = (1/R2.norm())*R2
Q3 = (1/R3.norm())*R3
Q = Matrix(BlockMatrix([Q1,Q2,Q3]))
SigA = diag(L0[0],L0[1]).row_join(Matrix([0,0]))
S1 = A*Q1
S2 = A*Q2
P1 = (1/S1.norm())*S1
P2 = (1/S2.norm())*S2
P = Matrix(BlockMatrix([P1,P2]))
P,SigA,Q,P*SigA*Q.T
```

Listing 4.6.1 *SymPy code for a singular value decomposition example*

1. *Do Exercise 8.6.9 in Nicholson: compute the SVD for the matrices*

$$\begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}.$$

Note that for these matrices, you will need to do some additional work to extend the p_i vectors to an orthonormal basis. You can adapt the code above, but you will have to think about how to implement additional code to construct any extra vectors you need.

2.   *Either by reading Nicholson or by searching online (or both), come up with a couple of answers to the question: "Why are people interested in the singular value decomposition?"*

3.   *(Optional) If you are interested, learn how to compute the SVD using tools from the NumPy and SciPy libraries instead.*

Chapter 5

Change of Basis

5.1 The matrix of a linear transformation

Recall from Example ?? in Chapter ?? that given any $m \times n$ matrix A, we can define the matrix transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$ by $T_A(\mathbf{x}) = A\mathbf{x}$, where we view $\mathbf{x} \in \mathbb{R}^n$ as an $n \times 1$ column vector.

Conversely, given any linear map $T : \mathbb{R}^n \to \mathbb{R}^m$, if we let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ denote the standard basis of $\mathbb{R}^n$, then the matrix

$$A = \begin{bmatrix} T(\mathbf{e}_1) & T(\mathbf{e}_2) & \cdots & T(\mathbf{e}_n) \end{bmatrix}$$

is such that $T = T_A$.

We have already discussed the fact that this idea generalizes: given a linear transformation $T : V \to W$, where V and W are finite-dimensional vector spaces, it is possible to represent T as a matrix transformation.

The representation depends on choices of bases for both V and W. Recall the definition of the coefficient isomorphism, from Definition ?? in Section ??. If $\dim V = n$ and $\dim W = m$, this gives us isomorphisms $C_B : V \to \mathbb{R}^n$ and $C_D : W \to \mathbb{R}^m$ depending on the choice of a basis B for V and a basis D for W. These isomorphisms define a matrix transformation $T_A : \mathbb{R}^n \to \mathbb{R}^m$ according to the diagram we gave in Figure ??.

We should stress one important point about the coefficient isomorphism, however. It depends on the choice of basis, but also on the order of the basis elements. Thus, we generally will work with an ordered basis in this chapter. That is, rather than simply thinking of our basis as a set, we will think of it as an ordered list. Order matters, since given a basis $B = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$, we rely on the fact that we can write any vector $\mathbf{v}$ uniquely as

$$\mathbf{v} = c_1 \mathbf{e}_1 + \cdots + c_n \mathbf{e}_n$$

in order to make the assignment $C_B(\mathbf{v}) = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$.

Exercise 5.1.1

Show that the coefficient isomorphism is, indeed, a linear isomorphism from V to $\mathbb{R}^n$.

Given $T : V \to W$ and coefficient isomorphisms $C_B : V \to \mathbb{R}^n, C_D : W \to \mathbb{R}^m$, the map $C_D T C_B^{-1} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear transformation, and the matrix of this transformation gives a representation of T. Explicitly, let $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ be an ordered basis for V, and let $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\}$ be an ordered basis for W. Since $T(\mathbf{v}_i) \in W$ for each $\mathbf{v}_i \in B$, there exist unique scalars a_{ij}, with $1 \le i \le m$ and $1 \le j \le n$ such that

$$T(\mathbf{v}_j) = a_{1j} \mathbf{w}_1 + a_{2j} \mathbf{w}_2 + \cdots + a_{mj} \mathbf{w}_m$$

for $j = 1, \ldots, n$. *This gives us the $m \times n$ matrix $A = [a_{ij}]$. Notice that the first column of A is $C_D(T(\mathbf{v}_1))$, the second column is $C_D(T(\mathbf{v}_2))$, and so on.*

Given $\mathbf{x} \in V$, write $\mathbf{x} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n$, so that $C_B(\mathbf{x}) = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$. Then

$$T_A(C_B(\mathbf{x})) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} a_{11}c_1 + a_{12}c_2 + \cdots + a_{1n}c_n \\ a_{21}c_1 + a_{22}c_2 + \cdots + a_{2n}c_n \\ \vdots \\ a_{m1}c_1 + a_{m2}c_2 + \cdots + a_{mn}c_n \end{bmatrix}.$$

On the other hand,

$$\begin{aligned} T(\mathbf{x}) &= T(c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n) \\ &= c_1T(\mathbf{v}_1) + \cdots + c_nT(\mathbf{v}_n) \\ &= c_1(a_{11}\mathbf{w}_1 + \cdots + a_{m1}\mathbf{w}_m) + \cdots c_n(a_{1n}\mathbf{w}_1 + \cdots + a_{mn}\mathbf{w}_m) \\ &= (c_1a_{11} + \cdots + c_na_{1n})\mathbf{w}_1 + \cdots + (c_1a_{m1} + \cdots + c_na_{mn})\mathbf{w}_m. \end{aligned}$$

Therefore,

$$C_D(T(\mathbf{x})) = \begin{bmatrix} c_1a_{11} + \cdots + c_na_{1n} \\ \vdots \\ c_1a_{m1} + \cdots + c_na_{mn} \end{bmatrix} = T_A(C_B(\mathbf{x})).$$

Thus, we see that $C_DT = T_AC_B$, or $T_A = C_DTC_B^{-1}$, as expected.

Definition 5.1.2 The matrix $M_{DB}(T)$ of a linear map.

Let V and W be finite-dimensional vector spaces, and let $T : V \to W$ be a linear map. Let $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ and $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\}$ be ordered bases for V and W, respectively. Then the **matrix $M_{DB}(T)$ of T** with respect to the bases B and D is defined by

$$M_{DB}(T) = \begin{bmatrix} C_D(T(\mathbf{v}_1)) & C_D(T(\mathbf{v}_2)) & \cdots & C_D(T(\mathbf{v}_n)) \end{bmatrix}.$$

In other words, $A = M_{DB}(T)$ is the unique $m \times n$ matrix such that $C_DT = T_AC_B$. This gives the defining property

$$C_D(T(\mathbf{v})) = M_{DB}(T)C_B(\mathbf{v}) \text{ for all } \mathbf{v} \in V,$$

as was demonstrated above.

> **Exercise 5.1.3**
>
> *Suppose $T : P_2(\mathbb{R}) \to \mathbb{R}^2$ is given by*
>
> $$T(a + bx + cx^2) = (a + c, 2b).$$
>
> *Compute the matrix of T with respect to the bases $B = \{1, 1 - x, (1 - x)^2\}$ of $P_2(\mathbb{R})$ and $D = \{(1, 0), (1, -1)\}$ of $\mathbb{R}^2$.*

When we compute the matrix of a transformation with respect to a non-standard basis, we don't have to worry about how to write vectors in the domain in terms of that basis. Instead, we simply plug the basis vectors into the transformation, and then determine how to write the output in terms of the basis of the codomain. However, if we want to use this matrix to compute values of $T : V \to W$, then we need a systematic way of writing elements of V in terms of the given basis.

Example 5.1.4 Working with the matrix of a transformation.

Let $T : P_2(\mathbb{R}) \to \mathbb{R}^2$ be a linear transformation whose matrix is given by

$$M(T) = \begin{bmatrix} 3 & 0 & 3 \\ -1 & -2 & 2 \end{bmatrix}$$

with respect to the ordered bases $B = \{1 + x, 2 - x, 2x + x^2\}$ of $P_2(\mathbb{R})$ and $D = \{(0, 1), (-1, 1)\}$ of $\mathbb{R}^2$. Find the value of $T(2 + 3x - 4x^2)$.

Solution. *We need to write the input $2 + 3x - 4x^2$ in terms of the basis B. This amounts to solving the system of equations given by*

$$a(1 + x) + b(2 - x) + c(2x + x^2) = 2 + 3x - 4x^2.$$

Of course, we can easily set up and solve this system, but let's try to be systematic, and obtain a more useful result for future problems. Since we can easily determine how to write any polynomial in terms of the standard basis $\{1, x, x^2\}$, it suffices to know how to write these three polynomials in terms of our basis.

At first, this seems like more work. After all, we now have three systems to solve:

$$a_1(x + 1) + b_1(2 - x) + c_1(2x + x^2) = 1$$
$$a_2(x + 1) + b_2(2 - x) + c_2(2x + x^2) = x$$
$$a_3(x + 1) + b_3(2 - x) + c_3(2x + x^2) = x^2.$$

However, all three systems have the same coefficient matrix, so we can solve them simultaneously, by adding three "constants" columns to our augmented matrix.

We get the matrix

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 0 & 1 & 0 & 0 \\ 1 & -1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

But this is exactly the augmented matrix we'd right down if we were trying to find the inverse of the matrix

$$P = \begin{bmatrix} 1 & 2 & 0 \\ 1 & -1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

whose columns are the coefficient representations of our given basis vectors in terms of the standard basis.

To compute P^{-1}, we use the computer:

```
from sympy import Matrix, init_printing
init_printing()
P = Matrix(3,3,[1,2,0,1,-1,2,0,0,1])
P**-1
```

\[\bbm \frac13\amp \frac23\amp -\frac43\\ \frac13\amp
    -\frac13 \amp \frac23\\0\amp 0\amp 1\ebm\]

Next, we find $M(T)P^{-1}$:

```
M = Matrix(2,3,[3,0,3,-1,-2,2])
v = Matrix(3,1,[2,3,-4])
M*P**-1
```

\[\bbm 1\amp 2\amp -1\\-1\amp 0\amp 2\ebm\]

   This matrix first converts the coefficient vector for a polynomial $p(x)$ with respect to the standard basis into the coefficient vector for our given basis B, and then multiplies by the matrix representing our transformation. The result will be the coefficient vector for $T(p(x))$ with respect to the basis D.

   The polynomial $p(x) = 2 + 3x - 4x^2$ has coefficient vector $\begin{bmatrix} 2 \\ 3 \\ -4 \end{bmatrix}$ with respect to the standard basis. We find that $M(T)P^{-1} \begin{bmatrix} 2 \\ 3 \\ -4 \end{bmatrix} = \begin{bmatrix} 12 \\ -10 \end{bmatrix}$:

```
M = Matrix(2,3,[3,0,3,-1,-2,2])
v = Matrix(3,1,[2,3,-4])
(M*P**-1)*v
```

\[\bbm 12\\-10\ebm\]

   The coefficients 12 and -10 are the coefficients of $T(p(x))$ with repsect to the basis D. Thus,

$$T(2 + 3x - 4x^2) = 12(0,1) - 10(-1,1) = (10,2).$$

Note that in the last step we gave the "simplified" answer $(10,2)$, which is simplified primarily in that it is expressed with respect to the standard basis.

   Note that we can also introduce the matrix $Q = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}$ whose columns are the coefficient vectors of the vectors in the basis D with respect to the standard basis. The effect of multiplying by Q is to convert from coefficients with respect to D into a coefficient vector with respect to the standard basis. We can then write a new matrix $\hat{M}(T) = QM(T)P^{-1}$; this new matrix is now the matrix representation of T with respect to the standard bases of $P_2(\mathbb{R})$ and $\mathbb{R}^2$.

```
Q = Matrix(2,2,[0,-1,1,1])
Q*M*P**-1
```

\[\bbm 1\amp 0\amp -2\\0\amp 2\amp 1\ebm\]

   We check that

$$\hat{M}(T) \begin{bmatrix} 2 \\ 3 \\ -4 \end{bmatrix} = \begin{bmatrix} 10 \\ 2 \end{bmatrix},$$

as before.

```
(Q*M*P**-1)*v
```

```
\[\bbm 10\\2 \ebm\]
```

We find that $\tilde{M}(T) = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 2 & 1 \end{bmatrix}$. This lets us determine that for a general polynomial $p(x) = a + bx + cx^2$,

$$\hat{M}(T) \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a - 2c \\ 2b + c \end{bmatrix},$$

and therefore, our original transformation must have been

$$T(a + bx + cx^2) = (a - 2c, 2b + c).$$

The previous example illustrated some important observations that are true in general. We won't give the general proof, but we sum up the results in a theorem.

Theorem 5.1.5

Suppose $T : V \to W$ is a linear transformation, and suppose $M_0 = M_{D_0 B_0}(T)$ is the matrix of T with respect to bases B_0 of V and D_0 of W. Let $B_1 = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ and $D_1 = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\}$ be any other choice of basis for V and W, respectively. Let

$$P = \begin{bmatrix} C_{B_0}(\mathbf{v}_1) & C_{B_0}(\mathbf{v}_2) & \cdots & C_{B_0}(\mathbf{v}_n) \end{bmatrix}$$
$$Q = \begin{bmatrix} C_{D_0}(\mathbf{w}_1) & C_{D_0}(\mathbf{w}_2) & \cdots & C_{DB_0}(\mathbf{w}_n) \end{bmatrix}$$

be matrices whose columns are the coefficient vectors of the vectors in B_1, D_1 with respect to B_0, D_0. Then the matrix of T with respect to the bases B_1 and D_1 is

$$M_{D_0 B_0}(T) = Q M_{D_1 B_1}(T) P^{-1}.$$

The relationship between the different maps is illustrated in Figure **??** below. In this figure, the maps $V \to V$ and $W \to W$ are the identity maps, corresponding to representing the same vector with respect to two different bases. The vertical arrows are the coefficient isomorphisms $C_{B_0}, C_{B_1}, C_{D_0}, C_{D_1}$.

In the HTML version of the book, you can click and drag to rotate the figure below.

Figure 5.1.6 Diagramming matrix of a transformation with respect to two different choices of basis

We generally apply Theorem **??** in the case that B_0, D_0 are the standard bases for V, W, since in this case, the matrices M_0, P, Q are easy to determine, and we can use a computer to calculate P^{-1} and the product $Q M_0 P^{-1}$.

Exercise 5.1.7

Suppose $T : M_{22}(\mathbb{R}) \to P_2(\mathbb{R})$ has the matrix

$$M_{DB}(T) = \begin{bmatrix} 2 & -1 & 0 & 3 \\ 0 & 4 & -5 & 1 \\ -1 & 0 & 3 & -2 \end{bmatrix}$$

with respect to the bases

$$B = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\}$$

of $M_{22}(\mathbb{R})$ and $D = \{1, x, x^2\}$ of $P_2(\mathbb{R})$. Determine a formula for T in terms of a general input $X = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

In textbooks such as Sheldon Axler's Linear Algebra Done Right that focus primarily on linear transformations, the above construction of the matrix of a transformation with respect to choices of bases can be used as a primary motivation for introducing matrices, and determining their algebraic properties. In particular, the rule for matrix multiplication, which can seem peculiar at first, can be seen as a consequence of the composition of linear maps.

> **Theorem 5.1.8**
>
> Let U, V, W be finite-dimensional vectors spaces, with ordered bases B_1, B_2, B_3, respectively. Let $T : U \to V$ and $S : V \to W$ be linear maps. Then
> $$M_{B_3 B_1}(ST) = M_{B_3 B_2}(S) M_{B_2 B_1}(T).$$

Proof. Let $\boldsymbol{x} \in U$. Then $C_{B_3}(ST(\boldsymbol{x})) = M_{B_3 B_1}(ST) C_{B_1}(\boldsymbol{x})$. On the other hand,

$$M_{B_3 B_2}(S) M_{B_2 B_1}(T) C_{B_1}(\boldsymbol{x}) = M_{B_3 B_2}(S)(C_{B_2}(T(\boldsymbol{x})))$$
$$= C_{B_3}(S(T(\boldsymbol{x}))) = C_{B_3}(ST(\boldsymbol{x})).$$

Since C_{B_3} is invertible, the result follows.   ∎

Being able to express a general linear transformation in terms of a matrix is useful, since questions about linear transformations can be converted into questions about matrices that we already know how to solve. In particular,

- $T : V \to W$ is an isomorphism if and only if $M_{DB}(T)$ is invertible for some (and hence, all) choice of bases B of V and D of W.

- The rank of T is equal to the rank of $M_{DB}(T)$ (and this does not depend on the choice of basis).

- The kernel of T is isomorphic to the nullspace of $M_{DB}(T)$.

Next, we will want to look at two topics in particular. First, if $T : V \to V$ is a linear operator, then it makes sense to consider the matrix $M_B(T) = M_{BB}(T)$ obtained by using the same basis for both domain and codomain. Second, we will want to know how this matrix changes if we change the choice of basis.

5.2 The matrix of a linear operator

Recall that a linear transformation $T : V \to V$ is referred to as a **linear operator**. Recall also that two matrices A and B are **similar** if there exists an invertible matrix P such that $B = PAP^{-1}$, and that similar matrices have a lot of properties in common. In particular, if A is similar to B, then A and B have the same trace, determinant, and eigenvalues. One way to understand this is the realization that two matrices are similar if they are representations of the same operator, with respect to different bases.

Since the domain and codomain of a linear operator are the same, we can consider the matrix $M_{DB}(T)$ where B and D are the same ordered basis. This leads to the next definition.

Definition 5.2.1

Let $T : V \to V$ be a linear operator, and let $B = \{b_1, b_2, \ldots, b_n\}$ be an ordered basis of V. The matrix $M_B(T) = M_{BB}(T)$ is called the B-**matrix** of T.

The following result collects several useful properties of the B-matrix of an operator. Most of these were already encountered for the matrix $M_{DB}(T)$ of a transformation, although not all were stated formally. (Formal statements can be found in the textbook by Nicholson.)

Theorem 5.2.2

Let $T : V \to V$ **be a linear operator, and let** $B = \{b_1, b_2, \ldots, b_n\}$ **be a basis for** V. **Then**

1. $C_B(T(v)) = M_B(T)C_B(v)$ **for all** $v \in V$.

2. **If** $S : V \to V$ **is another operator, then** $M_B(ST) = M_B(S)M_B(T)$.

3. T **is an isomorphism if and only if** $M_B(T)$ **is invertible.**

4. **If** T **is an isomorphism, then** $M_B(T^{-1}) = [M_B(T)]^{-1}$.

5. $M_B(T) = \begin{bmatrix} C_B(T(b_1)) & \cdots & C_B(T(b_n)) \end{bmatrix}$.

Example 5.2.3

Find the B-matrix of the operator $T : P_2(\mathbb{R}) \to P_2(\mathbb{R})$ given by $T(p(x)) = p(0)(1 + x^2) + p(1)x$, with respect to the ordered basis

$$B = \{1 - x, x + 3x^2, 2 - x^2\}.$$

Solution. *We compute*

$$T(1 - x) = 1(1 + x^2) + 0(x) = 1 + x^2$$
$$T(x + 3x^2) = 0(1 + x^2) + 4x = 4x$$
$$T(2 - x^2) = 2(1 + x^2) + 1(x) = 2 + x + 2x^2.$$

We now need to write each of these in terms of the basis B. We can do this by working out how to write each polynomial above in terms of B. Or we can be systematic.

Let $P = \begin{bmatrix} 1 & 0 & 2 \\ -1 & 1 & 0 \\ 0 & 3 & -1 \end{bmatrix}$ be the matrix whose columns are given by the coefficient representations of the polynomials in B with respect to the standard basis $B_0 = \{1, x, x^2\}$. For $T(1 - x) = 1 + x^2$ we need to solve the equation

$$a(1 - x) + b(x + 3x^2) + c(2 - x^2) = 1 + x^2$$

for scalars a, b, c. But this is equivalent to the system

$$a + 2c = 1$$

$$-a + b = 0$$
$$3b - c = 1,$$

which, in turn, is equivalent to the matrix equation

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 1 & 0 \\ 0 & 3 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix};$$

that is, $PC_B(1 + x^2) = C_{B_0}(1 + x^2)$. Thus,

$$C_B(1 + x^2) = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = P^{-1}C_{B_0}(1 + x^2) = P^{-1}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Similarly, $C_B(4x) = P^{-1}\begin{bmatrix} 0 \\ 4 \\ 0 \end{bmatrix} = P^{-1}C_{B_0}(4x)$, and $C_B(2 + x + 2x^2) =$

$$P^{-1}\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = P^{-1}C_{B_0}(2 + x + 2x^2).$$ *Using the computer, we find:*

```
from sympy import Matrix, init_printing
init_printing()
P = Matrix(3,3,[1,0,2,-1,1,0,0,3,-1])
M = Matrix(3,3,[1,0,2,0,4,1,1,0,2])
P**-1, P**-1*M
```

```
\[\left(\bbm 1/7\amp -6/7\amp 3/7\\1/7\amp 1/7\amp
    2/7\\3/7\amp 3/7\amp -1/7\ebm, \bbm 3/7\amp
    -24/7\amp 0\\3/7\amp 4/7\amp 1\\2/7\amp 12/7\amp
    1\ebm\right)\]
```

That is,

$$\begin{aligned} M_B(T) &= \begin{bmatrix} C_B(T(1 - x)) & C_B(T(x + 3x^2)) & C_B(T(2 - x^2)) \end{bmatrix} \\ &= \begin{bmatrix} P^{-1}C_{B_0}T(1 - x) & P^{-1}C_{B_0}T(x + 3x^2) & P^{-1}C_{B_0}(T(2 - x^2)) \end{bmatrix} \\ &= P^{-1}\begin{bmatrix} C_{B_0}(1 + x^2) & C_{B_0}(4x) & C_{B_0}(2 + x + 2x^2) \end{bmatrix} \\ &= \begin{bmatrix} 1/7 & -6/7 & 3/7 \\ 1/7 & 1/7 & 2/7 \\ 3/7 & 3/7 & -1/7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 4 & 1 \\ 1 & 0 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 3/7 & -24/7 & 0 \\ 3/7 & 4/7 & 1 \\ 2/7 & 12/7 & 1 \end{bmatrix}. \end{aligned}$$

Let's confirm that this works. Suppose we have

$$p(x) = C_B^{-1}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = a(1 - x) + b(x + 3x^2) + c(2 - x^2)$$

$$= (a + 2c) + (-a + b)x + (3b - c)x^2.$$

Then $T(p(x)) = (a + 2c)(1 + x^2) + (4b + c)x$, and we find

$$C_B(T(p(x))) = P^{-1} \begin{bmatrix} a + 2c \\ 4b + c \\ a + 2c \end{bmatrix} = \begin{bmatrix} \frac{3}{7}a - \frac{24}{7}b \\ \frac{3}{7}a + \frac{4}{7}b + c \\ \frac{2}{7}a + \frac{12}{7}b + c \end{bmatrix}.$$

On the other hand,

$$M_B(T) = \begin{bmatrix} 3/7 & -24/7 & 0 \\ 3/7 & 4/7 & 1 \\ 2/7 & 12/7 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \frac{3}{7}a - \frac{24}{7}b \\ \frac{3}{7}a + \frac{4}{7}b + c \\ \frac{2}{7}a + \frac{12}{7}b + c \end{bmatrix}.$$

The results agree, but possibly leave us a little confused.

In general, given an ordered basis $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ for a vector space V with standard basis $B_0 = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$, if we let

$$P = \begin{bmatrix} C_{B_0}(\mathbf{b}_1) & \cdots & C_{B_0}(\mathbf{b}_n) \end{bmatrix},$$

then

$$M_B(T) = \begin{bmatrix} C_B(T(\mathbf{b}_1)) & \cdots & C_B(T(\mathbf{b_n})) \end{bmatrix}$$
$$= P^{-1} \begin{bmatrix} C_{B_0}(T(\mathbf{b}_1)) & \cdots & C_{B_0}(T(\mathbf{b_n})) \end{bmatrix}.$$

As we saw above, this gives us the result, but doesn't shed much light on the problem, unless we have an easy way to write vectors in terms of the basis B. Let's revisit the problem. Instead of using the given basis B, let's use the standard basis $B_0 = \{1, x, x^2\}$. We quickly find

$$T(1) = 1 + x + x^2, T(x) = x, \text{ and } T(x^2) = x,$$

so with respect to the standard basis, $M_{B_0}(T) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. Now, recall that

$$M_B(T) = P^{-1} \begin{bmatrix} C_{B_0}T(1 - x) & C_{B_0}(T(x + 3x^2) & C_{B_0}(T(2 - x^2)) \end{bmatrix}$$

and note that for any polynomial $p(x)$, $C_{B_0}T(p(x)) = M_{B_0}C_{B_0}(p(x))$. But

$$\begin{bmatrix} C_{B_0}(1 - x) & C_{B_0}(x + 3x^2) & C_{B_0}(2 - x^2) \end{bmatrix} = P,$$

so we get

$$M_B(T) = P^{-1} \begin{bmatrix} C_{B_0}T(1 - x) & C_{B_0}(T(x + 3x^2) & C_{B_0}(T(2 - x^2)) \end{bmatrix}$$
$$= P^{-1} \begin{bmatrix} M_{B_0}C_{B_0}(1 - x) & M_{B_0}C_{B_0}(x + 3x^2) & M_{B_0}C_{B_0}(2 - x^2) \end{bmatrix}$$
$$= P^{-1}M_{B_0} \begin{bmatrix} C_{B_0}(1 - x) & C_{B_0}(x + 3x^2) & C_{B_0}(2 - x^2) \end{bmatrix}$$
$$= P^{-1}M_{B_0}P.$$

Now we have a much more efficient method for arriving at the matrix $M_B(T)$. The matrix $M_{B_0}(T)$ is easy to determine, the matrix P is easy to determine, and with the help of the computer, it's easy to compute $P^{-1}M_{B_0}P = M_B(T)$.

```
from sympy import Matrix, init_printing
init_printing()
M0 = Matrix(3,3,[1,0,0,1,1,1,1,0,0])
```

```
P**-1*M0*P
```

```
\[\bbm 3/7\amp -24/7\amp 0\\3/7\amp 4/7\amp 1\\2/7\amp
    12/7\amp 1\ebm\]
```

Exercise 5.2.4

Determine the matrix of the operator $T : \mathbb{R}^3 \to \mathbb{R}^3$ given by

$$T(x, y, z) = (3x - 2y + 4z, x - 5y, 2y - 7z)$$

with respect to the ordered basis

$$B = \{(1, 2, 0), (0, -1, 2), (1, 2, 1)\}.$$

(You may want to use the Sage cell below for computational assistance.)

The matrix P used in the above examples is known as a change matrix. If the columns of P are the coefficient vectors of $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\}$ with respect to another basis D, then we have

$$\begin{aligned}
P &= \begin{bmatrix} C_D(\boldsymbol{b}_1) & \cdots & C_D(\boldsymbol{b}_n) \end{bmatrix} \\
&= \begin{bmatrix} C_D(1_V(\boldsymbol{b}_1)) & \cdots & C_D(1_V(\boldsymbol{b}_n)) \end{bmatrix} \\
&= M_{DB}(1_V).
\end{aligned}$$

In other words, P is the matrix of the identity transformation $1_V : V \to V$, where we use the basis B for the domain, and the basis D for the codomain.

> **Definition 5.2.5**
>
> The **change matrix** with respect to ordered bases B, D of V is denoted $P_{D \leftarrow B}$, and defined by
> $$P_{D \leftarrow B} = M_{DB}(1_V).$$

> **Theorem 5.2.6**
>
> Let $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\}$ and D be two ordered bases of V. Then
> $$P_{D \leftarrow B} = \begin{bmatrix} C_D(\boldsymbol{b}_1) & \cdots & C_D(\boldsymbol{b}_n) \end{bmatrix},$$
> and satisfies $C_D(\boldsymbol{v}) = P_{D \leftarrow B} C_B(\boldsymbol{v})$ for all $\boldsymbol{v} \in V$.
>   The matrix $P_{D \leftarrow B}$ is invertible, and $(P_{D \leftarrow B})^{-1} = P_{B \leftarrow D}$. Moreover, if E is a third ordered basis, then
> $$P_{E \leftarrow D} P_{D \leftarrow B} = P_{E \leftarrow B}.$$

Example 5.2.7

Let $B = \{1, x, x^2\}$ and let $D = \{1 + x, x + x^2, 2 - 3x + x^2\}$ be ordered bases of $P_2(\mathbb{R})$. Find the change matrix $P_{D \leftarrow B}$.

Solution. *Finding this matrix requires us to first write the vectors in B in terms of the vectors in D. However, it's much easier to do this the other way*

around. We easily find

$$P_{B \leftarrow D} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & -3 \\ 0 & 1 & 1 \end{bmatrix},$$

and by Theorem ??, we have

$$P_{D \leftarrow B} = (P_{B \leftarrow D})^{-1} = \frac{1}{6} \begin{bmatrix} 4 & 2 & -2 \\ -1 & 1 & 5 \\ 1 & -1 & 1 \end{bmatrix}.$$

Note that the change matrix notation is useful for linear transformations between different vector spaces as well. Recall Theorem ??, which gave the result

$$M_{D_0 B_0}(T) = Q M_{D_1 B_1} P^{-1},$$

where (using our new notation) $P = P_{B_0 \leftarrow B_1}$ and $Q = P_{D_0 \leftarrow D_1}$. In this notation, we have

$$M_{D_0 B_0}(T) = P_{D_0 \leftarrow D_1} M_{D_1 B_1}(T) P_{B_1 \leftarrow B_0},$$

which seems more intiutive.

The above results give a straightforward procedure for determining the matrix of any operator, with respect to any basis, if we let D be the standard basis. The importance of these results is not just their computational simplicity, however. The most important outcome of the above is that if $M_B(T)$ and $M_D(T)$ give the matrix of T with respect to two different bases, then

$$M_B(T) = (P_{D \leftarrow B})^{-1} M_D(T) P_{D \leftarrow B},$$

so that the two matrices are similar.

Recall from Theorem ?? that similar matrices have the same determinant, trace, and eigenvalues. This means that we can unambiguously define the determinant and trace of an operator, and that we can compute eigenvalues of an operator using any matrix representation of that operator.

5.3 Direct Sums and Invariant Subspaces

Much of this section has been mentioned previously in the course (and these notes), but we will follow the organization of Nicholson's textbook, and reprise these concepts in more detail than previously.

5.3.1 Invariant subspaces

> **Definition 5.3.1**
>
> Given an operator $T : V \to V$, we say that a subspace $U \subseteq V$ is T-**invariant** if $T(\mathbf{u}) \in U$ for all $\mathbf{u} \in U$.

In other words, a subspace U is T-invariant if T does not map any vectors in U outside of U. Notice that if we shrink the domain of T to U, then we get an operator from U to U, since the image $T(U)$ is contained in U.

Given a basis $B = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ of U, note that U is T-invariant if and only if $T(\mathbf{u}_i) \in U$ for each $i = 1, 2, \ldots, k$.

For any operator $T : V \to V$, there are four subspaces that are always T-invariant:

$$\{\mathbf{0}\}, V, \ker T, \text{ and } \operatorname{im} T.$$

Of course, some of these subspaces might be the same; for example, if T is invertible, then $\ker T = \{\mathbf{0}\}$ and $\operatorname{im} T = V$. (We will skip the proof that $\ker T$ and $\operatorname{im} T$ are T-invariant, since this has been assigned as homework!)

> **Definition 5.3.2**
>
> Let $T : V \to V$ be a linear operator, and let U be a T-invariant subspace. The **restriction** of T to U, denoted $T|_U$, is the operator $T|_U : U \to U$ defined by $T|_U(\mathbf{u}) = T(\mathbf{u})$ for all $\mathbf{u} \in U$.

Notice that the restriction $T|_U$ is defined by the same "rule" as T, but its domain is the subspace U instead of the entire vector space V.

A lot can be learned by studying the restrictions of an operator to invariant subspaces. Indeed, the textbook by Axler does almost everything from this point of view. One reason to study invariant subspaces is that they allow us to put the matrix of T into simpler forms.

> **Theorem 5.3.3**
>
> Let $T : V \to V$ **be a linear operator, and let** U **be a** T-**invariant subspace. Let** $B_U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ **be a basis of** U**, and extend this to a basis**
>
> $$B = \{\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{w}_1, \ldots, \mathbf{w}_{n-k}\}$$
>
> **of** V**. Then the matrix** $M_B(T)$ **with respect to this basis has the block-triangular form**
>
> $$M_B(T) = \begin{bmatrix} M_{B_U}(T_U) & P \\ 0 & Q \end{bmatrix}$$
>
> **for some** $(n - k) \times (n - k)$ **matrix** Z**.**

Reducing a matrix to block triangular form is useful, because it simplifies computations such as determinants and eigenvalues (and determinants and eigenvalues are computationally expensive). In particular, if a matrix A has the block form

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ 0 & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn} \end{bmatrix},$$

where the diagonal blocks are square matrices, then $\det(A) = \det(A_{11}) \det(A_{22}) \cdots \det(A_{nn})$ and $c_A(x) = c_{A_{11}}(x) c_{A_{22}}(x) \cdots C_{A_{nn}}(x)$.

5.3.2 Eigenspaces

An important source of invariant subspaces is eigenspaces. Recall that for any real number λ, and any operator $T : V \to V$, we define

$$E_\lambda(T) = \ker(T - \lambda 1_V) = \{\mathbf{v} \in V \mid T(\mathbf{v}) = \lambda \mathbf{v}\}.$$

For most values of λ, we'll have $E_\lambda(T) = \{\mathbf{0}\}$. The values of λ for which $E_\lambda(T)$ is non-trivial are precisely the eigenvalues of T. Note that since similar matrices have the same characteristic polynomial any matrix representation $M_B(T)$ will have the

same eigenvalues. They do not generally have the same eigenspaces, but we do have the following.

Theorem 5.3.4

Let $T : V \to V$ be a linear operator. For any scalar λ, the eigenspace $E_\lambda(T)$ is T-invariant. Moreover, for any ordered basis B of V, the coefficient isomorphism $C_B : V \to \mathbb{R}^n$ induces an isomorphism

$$C_B|_{E_\lambda(T)} : E_\lambda(T) \to E_\lambda(M_B(T)).$$

5.3.3 Direct Sums

Recall that for any subspaces U, W of a vector space V, the sets

$$U + W = \{\boldsymbol{u} + \boldsymbol{w} \mid \boldsymbol{u} \in U \text{ and } \boldsymbol{w} \in W\}$$
$$U \cap W = \{\boldsymbol{v} \in V \mid \boldsymbol{v} \in U \text{ and } \boldsymbol{v} \in W\}$$

are subspaces of V. Saying that $\boldsymbol{v} \in U + W$ means that $\boldsymbol{v}$ can be written as a sum of a vector in U and a vector in W. However, this sum may not be unique. If $\boldsymbol{v} \in U \cap W$, $\boldsymbol{u} \in U$ and $\boldsymbol{w} \in W$, then we can write $(\boldsymbol{u}+\boldsymbol{v})+\boldsymbol{w} = \boldsymbol{u}+(\boldsymbol{v}+\boldsymbol{w})$, giving two different representations of a vector as an element of $U + W$.

Theorem 5.3.5

Let U and W be subspaces of a vector space V. Let $\boldsymbol{v} \in U + W$. Then there exist unique vectors $\boldsymbol{u} \in U, \boldsymbol{w} \in W$ such that $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}$ if and only if $U \cap W = \{\boldsymbol{0}\}$.

Proof. *Suppose $U + W$ has the property that if $\boldsymbol{v} \in U + W$, then there exist unique $\boldsymbol{u} \in U, \boldsymbol{w} \in W$ such that $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}$. Suppose $\boldsymbol{x} \in U \cap W$. Then $\boldsymbol{x} \in U$ and $\boldsymbol{x} \in W$, which implies that $-\boldsymbol{x} \in W$, since W is a subspace. Then we can write*

$$\boldsymbol{0} = \boldsymbol{x} + (-\boldsymbol{x}),$$

with $\boldsymbol{x} \in U$ and $-\boldsymbol{x} \in W$. But we also know that $\boldsymbol{0} \in U$ and $\boldsymbol{0} \in W$, so we can also write $\boldsymbol{0} = \boldsymbol{0} + \boldsymbol{0}$. Since we can only write $\boldsymbol{0}$ in one way as a sum of a vector in U and a vector in W, we must have $\boldsymbol{x} = \boldsymbol{0}$, showing that $U \cap W = \{\boldsymbol{0}\}$.

   Conversely, suppose that $U \cap W = \{\boldsymbol{0}\}$, and let $\boldsymbol{v} \in U + W$. Suppose that there exist vectors $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$ and $\boldsymbol{w}_1, \boldsymbol{w}_2 \in W$ such that $\boldsymbol{v} = \boldsymbol{u}_1 + \boldsymbol{w}_1 = \boldsymbol{u}_2 + \boldsymbol{w}_2$. But then $\boldsymbol{u}_1 - \boldsymbol{u}_2 = \boldsymbol{w}_2 - \boldsymbol{w}_1$, and since $\boldsymbol{u}_1 - \boldsymbol{u}_2 \in U$ and $\boldsymbol{w}_2 - \boldsymbol{w}_1 \in W$, we have $\boldsymbol{u}_1 - \boldsymbol{u}_2 = \boldsymbol{w}_2 - \boldsymbol{w}_1 \in U \cap W$. Since $U \cap W = \{\boldsymbol{0}\}$, we have $\boldsymbol{u}_1 - \boldsymbol{u}_2 = \boldsymbol{0}$ and $\boldsymbol{w}_2 - \boldsymbol{w}_1 = \boldsymbol{0}$, so $\boldsymbol{u}_1 = \boldsymbol{u}_2$ and $\boldsymbol{w}_1 = \boldsymbol{w}_2$. ∎

Definition 5.3.6

*We say that a sum $U + W$ is a **direct sum**, and write this sum as $U \oplus W$, if $U \cap W = \{\boldsymbol{0}\}$.*

---

   *Typically we are interested in the case that the two subspaces sum to V. If $V = U \oplus W$, we say that W is a **complement** of U, and that $U \oplus W$ is a direct sum decomposition of V. Of course, the orthogonal complement $U^\perp$ of a subspace U is a complement in this sense, if V is equipped with an inner product. (Without an inner product we have no concept of "orthogonal".) But even if we don't have an inner product, finding a complement is not too difficult, as the next example shows.*

Example 5.3.7 Finding a complement by extending a basis.

The easiest way to determine a direct sum decomposition (or equivalently, a complement) is through the use of a basis. Suppose U is a subspace of V with basis $\{e_1, e_2, \ldots, e_k\}$, and extend this to a basis

$$B = \{e_1, \ldots, e_k, e_{k+1}, \ldots, e_n\}$$

of V. Let $W = span\{e_{k+1}, \ldots, e_n\}$. Then clearly $U + W = V$, and $U \cap W = \{0\}$, since if $v \in U \cap W$, then $v \in U$ and $v \in W$, so we have

$$v = a_1 e_1 + \cdots + a_k e_k = b_1 e_{k+1} + \cdots + b_{n-k} e_n,$$

which gives

$$a_1 e_1 + \cdots + a_k e_k - b_1 e_{k+1} - \cdots - b_{n-k} e_n = 0,$$

so $a_1 = \cdots b_{n-k} = 0$ by the linear independence of B, showing that $v = 0$.
*   Conversely, if $V = U \oplus W$, and we have bases $\{u_1, u_2, \ldots, u_k\}$ of U and $\{v_1, v_2, \ldots, v_l\}$ of W, then*

$$B = \{u_1, \ldots, u_k, w_1, \ldots, w_l\}$$

is a basis for V. Indeed, B spans V, since every element of V can be written as $v = u + w$ with $u \in U, w \in W$. Independence follows by reversing the argument above: if

$$a_1 u_1 + \cdots + a_k u_k + b_1 w_1 + \cdots b_l w_l = 0$$

then $a_1 u_1 + \cdots + a_k u_k = -b_1 w_1 - \cdots - b_l w_l$, and equality is only possible if both sides belong to $U \cap W = \{0\}$. Since $\{u_1, u_2, \ldots, u_k\}$ is independent, the a_i have to be zero, and since $\{w_1, w_2, \ldots, w_l\}$ is independent, the b_j have to be zero.

The argument given in the second part of Example ?? has an immediate, but important consequence.

> **Theorem 5.3.8**
>
> **Suppose $V = U \oplus W$, where** *dim* $U = m$ **and** *dim* $W = n$. **Then V is finite-dimensional, and** *dim* $V = m + n$.

Example 5.3.9

Suppose $V = U \oplus W$, where U and W are T-invariant subspaces for some operator $T : V \to V$. Let $B_U = \{u_1, u_2, \ldots, u_m\}$ and let $B_W = \{w_1, w_2, \ldots, w_n\}$ be bases for U and W, respectively. Determine the matrix of T with respect to the basis $B = B_U \cup B_W$ of V.

Solution.   *Since we don't know the map T or anything about the bases B_U, B_W, we're looking for a fairly general statement here. Since U is T-invariant, we must have $T(u_i) \in U$ for each $i = 1, \ldots, m$. Similarly, $T(w_j) \in W$ for each $j = 1, \ldots, n$. This means that we have*

$$T(u_1) = a_{11} u_1 + \cdots + a_{m1} u_m + 0 w_1 + \cdots + 0 w_n$$

$$\vdots$$

$$T(\boldsymbol{u}_m) = a_{1m}\boldsymbol{u}_1 + \cdots + a_{mm}\boldsymbol{u}_m + 0\boldsymbol{w}_1 + \cdots + 0\boldsymbol{w}_n$$

$$T(\boldsymbol{w}_1) = 0\boldsymbol{u}_1 + \cdots + 0\boldsymbol{u}_m + b_{11}\boldsymbol{w}_1 + \cdots + b_{n1}\boldsymbol{w}_n$$

$$\vdots$$

$$T(\boldsymbol{w}_n) = 0\boldsymbol{u}_1 + \cdots + 0\boldsymbol{u}_m + b_{1n}\boldsymbol{w}_1 + \cdots + b_{nn}\boldsymbol{w}_n$$

for some scalars a_{ij}, b_{ij}. *If we set* $A = [a_{ij}]_{m \times m}$ *and* $B = [b_{ij}]_{n \times n}$, *then we have*

$$M_B(T) = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}.$$

Moreover, we can also see that $A = M_{B_U}(T|_U)$, *and* $B = M_{B_W}(T|_W)$.

5.4 Worksheet: generalized eigenvectors

Let V be a finite-dimensional vector space, and let $T : V \to V$ be a linear operator. Assume that T has all real eigenvalues (alternatively, assume we're working over the complex numbers). Let A be the matrix of T with respect to some standard basis B_0 of V.

Our goal will be to replace the basis B_0 with a basis B such that the matrix of T with respect to B is as simple as possible. (Where we agree that the "simplest" possible matrix would be diagonal.)

Recall the following results that we've observed so far:

- The characteristic polynomial $c_T(x)$ of T does not depend on the choice of basis.

- The eigenvalues of T are the roots of this polynomial.

- The eigenspaces $E_\lambda(T)$ are T-invariant subspaces of V.

- The matrix A can be diagonalized if and only if there is a basis of V consisting of eigenvectors of T.

- Suppose
$$c_T(x) = (x - \lambda_1)^{m_1} (x - \lambda_2)^{m_2} \cdots (x - \lambda_k)^{m_k}.$$
Then A can be diagonalized if and only if $\dim E_{\lambda_i}(T) = m_i$ for each $i = 1, \ldots, k$.

In the case where A can be diagonalized, we have the direct sum decomposition
$$V = E_{\lambda_1}(T) \oplus E_{\lambda_2}(T) \oplus \cdots \oplus E_{\lambda_k}(T).$$

The question is: what do we do if there aren't enough eigenvectors to form a basis of V? When that happens, the direct sum of all the eigenspaces will not give us all of V.

The idea: replace $E_{\lambda_j}(T)$ with a **generalized eigenspace** $G_{\lambda_j}(T)$ whose dimension is m_i.

Our candidate: instead of $E_\lambda(T) = ker(T - \lambda I)$, we use $G_\lambda(T) = ker((T - \lambda I)^m)$, where m is the multiplicity of λ.

1. Recall that in class we proved that $\ker(T)$ and $\text{im}(T)$ are T-invariant subspaces. Let $p(x)$ be any polynomial, and prove that $\ker(p(T))$ and $\text{im}(p(T))$ are also T-invariant.
    Hint: first show that $p(T)T = Tp(T)$ for any polynomial T.

Applying the result of Problem 1 to the polynomial $p(x) = (x - \lambda)^m$ shows that $G_\lambda(T)$ is T-invariant. It is possible to show that $\dim G_\lambda(T) = m$ but I won't ask you to do that. (A proof is in Nicholson if you really want to see it.)
    Instead, we will try to understand what's going on by exploring an example.
    Consider the following matrix.

```
from sympy import *
init_printing()
A=Matrix([[2,0,0,1,0],[-1,0,1,2,3],[0,1,2,0,-1],[-2,-3,2,5,3],[0,-1,0,1,4]])
A
```

2. Find (and factor) the characteristic polynomial of A. For the commands you might need, *refer to the textbook*[1].

---

3.   *Find the eigenvectors. What are the dimensions of the eigenspaces? Based on this observation, can A be diagonalized?*

4.   *Prove that for any $n \times n$ matrix A, we have*

$$\{0\} \subseteq null(A) \subseteq null(A^2) \subseteq \cdots \subseteq null(A^n).$$

    It turns out that at some point, the null spaces stabilize. If $\text{null}(A^k) = \text{null}\,A^{k+1}$ *for some* k, *then* $\text{null}(A^k) = \text{null}(A^{k+l})$ *for all* $l \geq 0$.

5.   *For each eigenvalue found in* Worksheet Exercise 5.4.2, *compute the nullspace of* $A - \lambda I$, $(A - \lambda I)^2$, $(A - \lambda I)^3$, *etc. until you find two consecutive nullspaces that are the same.*

    By Worksheet Exercise 5.4.4, *any vector in* $\text{null}(A - \lambda I)^m$ *will also be a vector in* $\text{null}(A - \lambda I)^{m+1}$. *In particular, at each step, we can find a basis for* $\text{null}(A - \lambda I)^m$ *that includes the basis for* $\text{null}(A - \lambda I)^{m-1}$.

    For each eigenvalue found in Worksheet Exercise 5.4.2, *determine such a basis for the corresponding generalized eigenspace. You will want to list your vectors so that the vectors from the basis of the nullspace for* $A - \lambda I$ *come first, then the vectors for the basis of the nullspace for* $(A - \lambda I)^2$, *and so on.*

6.   *Finally, let's see how all of this works. Let* P *be the matrix whose columns consist of the vectors found in Problem 4. What do you get when you compute the matrix* $P^{-1}AP$?

5.5 Generalized eigenspaces

*Example Example **??** showed us that if $V = U \oplus W$, where U and W are T-invariant, then the matrix $M_B(T)$ has block diagonal form $\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$, as long as the basis B is the union of bases of U and W.*

We want to take this idea further. If $V = U_1 \oplus U_2 \oplus \cdots \oplus U_k$, where each subspace U_j is T-invariant, then with respect to a basis B consisting of basis vectors for each subspace, we will have

$$M_B(T) = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_k \end{bmatrix},$$

where each A_j is the matrix of $T|_{U_j}$ with respect to some basis of U_j.

Our goal moving forward is twofold: one, to make the blocks as small as possible, so that $M_B(T)$ is as close to diagonal as possible, and two, to make the blocks as simple as possible. Of course, if T is diagonalizable, then we can get all blocks down to size 1×1, but this is not always possible.

*Recall from Section **??** that if the characteristic polynomial of T (or equivalently, any matrix representation A of T) is*

$$c_T(x) = (x - \lambda_1)^{m_1}(x - \lambda_2)^{m_2} \cdots (x - \lambda_k)^{m_k},$$

then $\dim E_{\lambda_j}(T) \le m_j$ for each $j = 1, \ldots, k$, and T is diagonalizable if and only if we have equality for each j. (This guarantees that we have sufficiently many independent eigenvectors to form a basis of V.)

Since eigenspaces are T-invariant, we see that being able to diagonalize T is equivalent to having the direct sum decomposition

$$V = E_{\lambda_1}(T) \oplus E_{\lambda_2}(T) \oplus \cdots \oplus E_{\lambda_k}(T).$$

If T cannot be diagonalized, it's because we came up short on the number of eigenvectors, and the direct sum of all eigenspaces only produces some subspace of V of lower dimension. We now consider how one might enlarge a set of independent eigenvectors in some standard, and ideally optimal, way.

First, we note that for any operator T, the restriction of T to $\ker T$ is the zero operator, since by definition, $T(\mathbf{v}) = \mathbf{0}$ for all $\mathbf{v} \in \ker T$. Since we define $E_\lambda(T) = \ker(T - \lambda I)$, it follows that $T - \lambda I$ restricts to the zero operator on the eigenspace $E_\lambda(T)$. The idea is to relax the condition "identically zero" to something that will allow us to potentially enlarge some of our eigenspaces, so that we end up with enough vectors to span V.

It turns out that the correct replacement for "identically zero" is "nilpotent". What we would like to find is some subspace $G_\lambda(T)$ such that the restriction of $T - \lambda I$ to $G_\lambda(T)$ will be nilpotent. (Recall that this means $(T - \lambda I)^k = 0$ for some integer k when restricted to $G_\lambda(T)$.) The only problem is that we don't (yet) know what this subspace should be. To figure it out, we rely on some ideas you may have explored in your last assignment.

Theorem 5.5.1

Let $T : V \to V$ **be a linear operator. Then:**

1. $\{\mathbf{0}\} \subseteq \ker T \subseteq \ker T^2 \subseteq \cdots \subseteq \ker T^k \subseteq \cdots$

2. If $\ker T^{k+1} = \ker T^k$ **for some** k, **then** $\ker T^{k+m} = \ker T^k$ **for all** $m \geq 0$.

3. **If** $n = \dim V$, **then** $\ker T^{n+1} = \ker T^n$.

In other words, for any operator T, *the kernels of successive powers of* T *can get bigger, but the moment the kernel doesn't change for the next highest power, it stops changing for all further powers of* T. *That is, we have a sequence of kernels of strictly greater dimension until we reach a maximum, at which point the kernels stop growing. And of course, the maximum dimension cannot be more than the dimension of* V.

Definition 5.5.2

Let $T : V \to V$ *be a linear operator, and let* λ *be an eigenvalue of* T. *The **generalized eigenspace** of* T *associated to the eigenvalue* λ *is denoted* $G_\lambda(T)$, *and defined as*

$$G_\lambda(T) = \ker(T - \lambda I)^n,$$

where $n = \dim V$.

Some remarks are in order. First, we can actually define $G_\lambda(T)$ *for any scalar* λ. *But this space will be trivial if* λ *is not an eigenvalue. Second, it is possible to show (although we will not do so here) that if* λ *is an eigenvalue with multiplicity* m, *then* $G_\lambda(T) = \ker(T - \lambda I)^m$. *(The kernel will usually have stopped growing well before we hit* $n = \dim V$, *but we know they're all eventually equal, so using* n *guarantees we have everything).*

We will not prove it here (see Nicholson, or Axler), but the advantage of using generalized eigenspaces is that they're just big enough to cover all of V.

Theorem 5.5.3

Let V **be a complex vector space, and let** $T : V \to V$ **be a linear operator. (We can take** V **to be real if we assume that** T **has all real eigenvalues.) Let** $\lambda_1, \ldots, \lambda_k$ **be the distinct eigenvalues of** T. **Then each generalized eigenspace** $G_{\lambda_j}(T)$ **is** T**-invariant, and we have the direct sum decomposition**

$$V = G_{\lambda_1}(T) \oplus G_{\lambda_2}(T) \oplus \cdots \oplus G_{\lambda_k}(T).$$

For each eigenvalue λ_j *of* T, *let* l_j *denote the smallest integer power such that* $G_{\lambda_j}(T) = (T - \lambda_j I)^{l_j}$. *Then certainly we have* $l_j \leq m_j$ *for each* j. *(Note also that if* $l_j = 1$, *then* $G_{\lambda_j}(T) = E_{\lambda_j}(T)$.)*

The polynomial $m_T(x) = (x - \lambda_1)^{l_1}(x - \lambda_2)^{l_2} \cdots (x - \lambda_k)^{l_k}$ *is the polynomial of smallest degree such that* $m_T(T) = 0$. *The polynomial* $m_T(x)$ *is called the **minimal polynomial** of* T. *Note that* T *is diagonalizable if and only if the minimal polynomial of* T *has no repeated roots.*

*In Section **??**, we'll explore a systematic method for determining the generalized eigenspaces of a matrix, and in particular, for computing a basis for each generalized eigenspace, with respect to which the corresponding block in the block-diagonal form is especially simple.*

5.6 Jordan Canonical Form

The results of Theorem ?? and Theorem ?? tell us that for an eigenvalue λ of $T : V \to V$ with multiplicity m, we have a sequence of subspace inclusions

$$E_\lambda(T) = \text{ker}(T - \lambda I) \subseteq \text{ker}(T - \lambda I)^2 \subseteq \cdots \subseteq \text{ker}(T - \lambda I)^m = G_\lambda(T).$$

Not all subspaces in this sequence are necessarily distinct. Indeed, it is entirely possible that dim $E_\lambda(T) = m$, in which case $E_\lambda(T) = G_\lambda(T)$. In geeral there will be some $l \leq m$ such that $\text{ker}(T - \lambda I)^l = G_\lambda(T)$.

Our goal in this section is to determine a basis for $G_\lambda(T)$ in a standard way. We begin with a couple of important results, which we state without proof. The first can be found in Axler's book; the second in Nicholson's.

Theorem 5.6.1

Suppose V is a complex vector space, and $T : V \to V$ is a linear operator. Let $\lambda_1, \ldots, \lambda_k$ denote the distinct eigenvalues of T. (We can assume V is real if we also assume that all eigenvalues of V are real.) Then:

1. **Generalized eigenvectors corresponding to distinct eigenvalues are linearly independent.**

2. **$V = G_{\lambda_1}(T) \oplus G_{\lambda_2}(T) \oplus \cdots \oplus G_{\lambda_k}(T)$**

3. **Each generalize eigenspace $G_{\lambda_j}(T)$ is T-invariant.**

4. **Each restriction $(T - \lambda_j)|_{G_{\lambda_j}(T)}$ is nilpotent.**

Theorem 5.6.2

Let $T : V \to V$ be a linear operator. If the characteristic polynomial of T is given by

$$c_T(x) = (x - \lambda_1)^{m_1}(x - \lambda_2)^{m_2} \cdots (x - \lambda_k)^{m_k},$$

then *dim* $G_{\lambda_j}(T) = m_j$ for each $j = 1, \ldots, k$.

Moreover, if we let $B = B_1 \cup B_2 \cup \cdots \cup B_k$, where B_j is any basis for $G_{\lambda_j}(T)$ for $j = 1, \ldots, k$, then B is a basis for V (this follows immediately from Theorem ??) and the matrix of T with respect to this basis has the block-diagonal form

$$M_B(T) = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_k \end{bmatrix},$$

where each A_j has size $m_j \times m_j$.

A few remarks are called for here.

- *One of the ways to see that dim $G_{\lambda_j}(T) = m_j$ is to consider $(M_B(T) - \lambda_j I_n)^{m_j}$. This will have the form diag$(U_1^{m_j}, U_2^{m_j}, \ldots, U_k^{m_j})$, where U_i is the matrix of $(T - \lambda_j)^{m_j}$, restricted to $G_{\lambda_i}(T)$. If $i \neq j$, $T - \lambda_j I$ restricts to an invertible operator on $G_{\lambda_i}(T)$, but its restriction to $G_{\lambda_j}(T)$ is nilpotent, by Theorem ??. So U_j is nilpotent (with $U_j^{m_j} = 0$), and has size $m_j \times m_j$, while U_i is invertible*

if $i \neq j$. The matrix $(M_B(T) - \lambda_j I)^{m_j}$ thus ends up with a $m_j \times m_j$ block of zeros, so $\dim \ker(T - \lambda_j I)^{m_j} = m_j$.

- If the previous point wasn't clear, note that with an appropriate choice of basis, the block A_i in Theorem **??** has the form

$$A_i = \begin{bmatrix} \lambda_i & * & \cdots & * \\ 0 & \lambda_i & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_i \end{bmatrix}.$$

  Thus, $M_B(T) - \lambda_j I$ will have blocks that are upper triangular, with diagonal entries $\lambda_i - \lambda_j \neq 0$ when $i \neq j$, but when $i = j$ we get a matrix that is strictly upper triangular, and therefore nilpotent, since its diagonal entries will be $\lambda_j - \lambda_j = 0$.

- if l_j is the least integer such that $\ker(A - \lambda_j I)^{l_j} = G_{\lambda_j}(T)$, then it is possible to choose the basis of $G_{\lambda_j}(T)$ so that A_j is itself block-diagonal, with the largest block having size $l_j \times l_j$. The remainder of this section is devoted to determining how to choose such a basis.

The basic principle for choosing a basis for each generalized eigenspace is as follows. We know that $E_\lambda(T) \subseteq G_\lambda(T)$ for each eigenvalue λ. So we start with a basis for $E_\lambda(T)$, by finding eigenvectors as usual. If $\ker(T - \lambda I)^2 = \ker(T - \lambda I)$, then we're done: $E_\lambda(T) = G_\lambda(T)$. Otherwise, we enlarge the basis for $E_\lambda(T)$ to a basis of $\ker T(-\lambda I)^2$. If $\ker T(-\lambda I)^3 = \ker(T - \lambda I)^2$, then we're done, and $G_\lambda(T) = \ker(T - \lambda I)^2$. If not, we enlarge our existing basis to a basis of $\ker(T - \lambda I)^3$. We continue this process until we reach some power l such that $\ker(T - \lambda I)^l = \ker(T - \lambda I)^{l+1}$. (This is guaranteed to happen by Theorem **??**.) We then conclude that $G_\lambda(T) = \ker(T - \lambda I)^l$.

The above produces a basis for $G_\lambda(T)$, but we want what is, in some sense, the "best" basis. For our purposes, the best basis is the one in which the matrix of T restricted to each generalized eigenspace is block diagonal, where each block is a Jordan block.

Definition 5.6.3

Let λ be a scalar. A **Jordan block** is an $m \times m$ matrix of the form

$$J(m, \lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & 0 & \cdots & \lambda \end{bmatrix}.$$

That is $J(m, \lambda)$ has each diagonal entry equal to λ, and each "superdiagonal" entry (those just above the diagonal) equal to 1, with all other entries equal to zero.

Example 5.6.4

The following are examples of Jordan blocks:

$$J(2,4) = \begin{bmatrix} 4 & 1 \\ 0 & 4 \end{bmatrix}, J(3, \sqrt{2}) = \begin{bmatrix} \sqrt{2} & 1 & 0 \\ 0 & \sqrt{2} & 1 \\ 0 & 0 & \sqrt{2} \end{bmatrix}, J(4, 2i) = \begin{bmatrix} 2i & 1 & 0 & 0 \\ 0 & 2i & 1 & 0 \\ 0 & 0 & 2i & 1 \\ 0 & 0 & 0 & 2i \end{bmatrix}.$$

Insight 5.6.5 Finding a chain basis.

A Jordan block corresponds to basis vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ with the following properties:

$$T(\mathbf{v}_1) = \lambda \mathbf{v}_1$$
$$T(\mathbf{v}_2) = \mathbf{v}_1 + \lambda \mathbf{v}_2$$
$$T(\mathbf{v}_3) = \mathbf{v}_2 + \lambda \mathbf{v}_3,$$

and so on. Notice that $\mathbf{v}_1$ is an eigenvector, and for each $j = 2, \ldots, m$,

$$(T - \lambda I)\mathbf{v}_j = \mathbf{v}_{j-1}.$$

Notice also that if we set $N = T - \lambda I$, then

$$\mathbf{v}_1 = N\mathbf{v}_2, \mathbf{v}_2 = N\mathbf{v}_3, \ldots, \mathbf{v}_{m-1} = N\mathbf{v}_m$$

so our basis for $G_\lambda(T)$ is of the form

$$\mathbf{v}, N\mathbf{v}, N^2\mathbf{v}, \ldots, N^{m-1}\mathbf{v},$$

where $\mathbf{v} = \mathbf{v}_m$, and $\mathbf{v}_1 = N^{m-1}\mathbf{v}$ is an eigenvector. (Note that $N^m \mathbf{v} = (T - \lambda I)\mathbf{v}_1 = \mathbf{0}$, and indeed $N^m \mathbf{v}_j = \mathbf{0}$ for each $j = 1, \ldots, m$.) Such a basis is known as a **chain basis**.

If $\dim E_\lambda(T) > 1$ we might have to repeat this process for each eigenvector in a basis for the eigenspace. The full matrix of T might have several Jordan blocks of possibly different sizes for each eigenvalue.

Example 5.6.6

Determine a Jordan basis for the operator $T : \mathbb{R}^5 \to \mathbb{R}^5$ whose matrix with respect to the standard basis is given by

$$A = \begin{bmatrix} 7 & 1 & -3 & 2 & 1 \\ -6 & 2 & 4 & -2 & -2 \\ 0 & 1 & 3 & 1 & -1 \\ -8 & -1 & 6 & 0 & -3 \\ -4 & 0 & 3 & -1 & 1 \end{bmatrix}$$

Solution. First, we need the characteristic polynomial.

```
from sympy import Matrix,init_printing,factor
init_printing()
A = Matrix([[7,1,-3,2,1],
            [-6,2,4,-2,-2],
            [0,1,3,1,-1],
            [-8,-1,6,0,-3],
            [-4,0,3,-1,1]])
p = A.charpoly().as_expr()
factor(p)
```

\[(\lambda -3)^3(\lambda -2)^2\]

The characteristic polynomial of A is given by

$$c_A(x) = (x-2)^2(x-3)^3.$$

We thus have two eigenvalues: 2, of multiplicity 2, and 3, of multiplicity 3. We next find the $E_2(A)$ eigenspace.

```
from sympy import eye
N2 = A-2*eye(5)
E2 = N2.nullspace()
E2
```

\[\bbm -1\\0\\-1\\1\\0\ebm\]

The computer gives us

$$E_2(A) = null(A - 2I) = span\{\mathbf{x}_1\}, \text{ where } \mathbf{x}_1 = \begin{bmatrix} -1 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix},$$

so we have only one independent eigenvector, which means that $G_2(A) = null(A - 2I)^2$.

Following Insight **??**, we extend $\{\mathbf{x}_1\}$ to a basis of $G_2(A)$ by solving the system

$$(A - 2I)\mathbf{x} = \mathbf{x}_1.$$

```
B2 = N2.col_insert(5,E2[0])
B2.rref()
```

\[\left(\bbm 1\amp 0\amp 0 \amp 1\amp 0\amp 0\\0\amp
    1\amp 0\amp 0\amp 0\amp -1\\0\amp 0\amp 1\amp
    1\amp 0\amp 0\\
0\amp 0\amp 0\amp 0\amp 1\amp 0\\0\amp 0\amp 0\amp
    0\amp 0\amp 0\ebm,(0,1,2,4)\right)\]

Using the results above from the computer (or Gaussian elimination), we

find a general solution

$$\mathbf{x} = \begin{bmatrix} -t \\ -1 \\ -t \\ t \\ 0 \end{bmatrix} = t \begin{bmatrix} -1 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Note that our solution is of the form $\mathbf{x} = t\mathbf{x}_1 + \mathbf{x}_2$. We set $t = 0$, and get $\mathbf{x}_2 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \end{bmatrix}^T$.

Next, we consider the eigenvalue $\lambda = 3$. The computer gives us the following:

```
N3 = A-3*eye(5)
E3 = N3.nullspace()
E3
```

$$\left[\bbm \frac12\\-1\\1\\1\\0\ebm, \bbm -\frac12\\1\\0\\0\\1\ebm\right]$$

Rescaling to remove fractions, we find

$$E_3(A) = null(A - 3I) = span\{\mathbf{y}_1, \mathbf{y}_2\}, \text{ where } \mathbf{y}_1 = \begin{bmatrix} 1 \\ -2 \\ 2 \\ 2 \\ 0 \end{bmatrix}, \mathbf{y}_2 = \begin{bmatrix} -1 \\ 2 \\ 0 \\ 0 \\ 2 \end{bmatrix}.$$

Again, we're one eigenvector short of the multiplicity, so we need to consider $G_3(A) = null(A - 3I)^3$.

In the next cell, note that we doubled the eigenvectors in E3 to avoid fractions. To follow the solution in our example, we append 2*E3[0], and reduce the resulting matrix. You should find that using the eigenvector $\mathbf{y}_1$ corresponding to E3[0] leads to an inconsistent system. Once you confirm this, replace E3[0] with E3[1] and re-run the cell to see that we get an inconsistent system using $\mathbf{y}_2$ as well!

```
B3 = N3.col_insert(5,2*E3[0])
B3.rref()
```

$$\left(\bbm 1\amp 0\amp 0\amp -\frac12\amp \frac12\amp 0\\0\amp 1\amp 0\amp 1\amp -1\amp 0\\0\amp 0\amp 1\amp -1\amp 0\amp 0\\ 0\amp 0\amp 0\amp 0\amp 0\amp 1\\0\amp 0\amp 0\amp 0\amp 0\amp 0\ebm, (0,1,2,5)\right)$$

The systems $(A-3I)\mathbf{y} = \mathbf{y}_1$ and $(A-3I)\mathbf{y} = \mathbf{y}_2$ are both inconsistent, but we can salvage the situation by replacing the eigenvector $\mathbf{y}_2$ by some linear combination $\mathbf{z}_2 = a\mathbf{y}_1 + b\mathbf{y}_2$. We row-reduce, and look for values of a and b that give a consistent system.

The rref command takes things a bit farther than we'd like, so we use the command echelon_form() instead. Note that if $a = b$, the system is inconsistent.

```
from sympy import Symbol
a = Symbol('a')
b = Symbol('b')
C3 = N3.col_insert(5,a*E3[0]+b*E3[1])
C3.echelon_form()
```

$$\bbm 4\amp 1\amp -3\amp 2\amp 1\amp \frac{a}{2}-\frac{b}{2}\\ 0\amp 2\amp -2\amp 4\amp -2\amp -a+b\\ 0\amp 0\amp 2\amp -2\amp 0\amp 3a-b\\ 0\amp 0\amp 0\amp 0\amp 0\amp 16a-16b\\ 0\amp 0\amp 0\amp 0\amp 0\amp 0\ebm$$

We find that $a = b$ does the job, so we set

$$\boldsymbol{z}_2 = \boldsymbol{y}_1 + \boldsymbol{y}_2 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 2 \\ 2 \end{bmatrix}.$$

```
D3 = N3.col_insert(5,E3[0]+E3[1])
D3.rref()
```

$$\left(\bbm 1\amp 0\amp 0\amp -\frac12\amp \frac12\amp \frac12\\ 0\amp 1\amp 0\amp 1\amp -1 \amp 1\\0\amp 0\amp 1\amp -1\amp 0\amp 1\\ 0\amp 0\amp 0\amp 0\amp 0\amp 0\\0\amp 0\amp 0\amp 0\amp 0\amp 0\ebm,(0,1,2)\right)$$

Solving the system $(A - 3I)\boldsymbol{z} = \boldsymbol{y}_1 + \boldsymbol{y}_2$, using the code above, we find

$$\boldsymbol{z} = \begin{bmatrix} \frac{1}{2} + \frac{1}{2}s - \frac{1}{2}t \\ 1 - s + t \\ 1 + s \\ s \\ t \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2} \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + s \begin{bmatrix} \frac{1}{2} \\ -1 \\ 1 \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} -\frac{1}{2} \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2} \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \frac{s}{2}\boldsymbol{y}_1 + \frac{t}{2}\boldsymbol{y}_2.$$

We let $\mathbf{z}_3 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \end{bmatrix}$, and check that

$$A\mathbf{z}_3 = 3\mathbf{z}_3 + \mathbf{z}_2,$$

as required:

```
Z3 = Matrix(5,1,[1,2,2,0,0])
A*Z3-3*Z3-2*(E3[0]+E3[1])
```

    \[\bbm 0\\0\\0\\0\\0\ebm\]

This gives us the basis $B = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{z}_2, \mathbf{z}_3\}$ for $\mathbb{R}^5$, and with respect to this basis, we have the Jordan canonical form

$$M_B(T) = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}.$$

Now that we've done all the work required for Example **??**, we should confess that there was an easier way all along:

```
A.jordan_form()
```

    \[\left(\bbm 1\amp 0\amp 0\amp \frac12\amp \frac12\\
        0\amp 1\amp 0\amp 1\amp -1\\
    1\amp 0\amp 1\amp 1\amp 1\\-1\amp 0\amp 1\amp 0\amp
        1\\0\amp 0\amp 1\amp 0\amp 0\ebm,
    \bbm 2\amp 1\amp 0\amp 0\amp 0\\0\amp 2\amp 0\amp
        0\amp 0\\
    0\amp 0\amp 3\amp 1\amp 0\\0\amp 0\amp 0\amp 3\amp
        0\\0\amp 0\amp 0\amp 0\amp 3\ebm\right)\]

The `jordan_form()` command returns a pair P, J, where J is the Jordan canonical form of A, and P is an invertible matrix such that $P^{-1}AP = J$. You might find that the computer's answer is not quite the same as ours. This is because the Jordan canonical form is only unique up to permutation of the Jordan blocks. Changing the order of the blocks amounts to changing the order of the columns of P, which are given by a basis of (generalized eigenvectors).

Exercise 5.6.7

Determine a Jordan basis for the linear operator $T : \mathbb{R}^4 \to \mathbb{R}^4$ given by

$$T(w, x, y, z) = (w + x, x, -x + 2y, w - x + y + z).$$

A code cell is given below in case you want to try performing the operations demonstrated in Example **??**.

One final note: we mentioned above that the minimal polynomial of an operator

has the form

$$m_T(x) = (x - \lambda_1)^{l_1}(x - \lambda_2)^{l_2} \cdots (x - \lambda_k)^{l_k},$$

where for each $j = 1, 2, \ldots, k$, l_j is the size of the largest Jordan block correspond-ing to λ_j. Knowing the minimal polynomial therefore tells as a lot about the Jordan canonical form, but not everything. Of course, if $l_j = 1$ for all j, then our operator can be diagaonalized. If $\dim V \leq 4$, the minimal polynomial tells us everything, except for the order of the Jordan blocks.

*In Exercise **??**, the minimal polynomial is $m_T(x) = (x - 1)^3(x - 2)$, the same as the characteristic polynomial. If we knew this in advance, then the only possible Jordan canonical forms would be*

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad or \quad \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

If instead the minimal polynomial had turned out to be $(x-1)^2(x-2)$ (with the same characteristic polynomial), then, up to permutation of the Jordan blocks, our Jordan canonical form would be

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}.$$

However, once we hit matrices of size 5×5 or larger, some ambiguity creeps in. For example, suppose $c_A(x) = (x - 2)^5$ with $m_A(x) = (x - 2)^2$. Then the largest Jordan block is 2×2, but we could have two 2×2 blocks and a 1×1, or three 1×1 blocks and one 2×2.

Appendix A

Computational Tools

A.1 Jupyter

*The first thing you need to know about doing linear algebra in Python is how to access a Python environment. Fortunately, you do not need to install any software for this. The University of Lethbridge has access to the **Syzygy Jupyter Hub** service, provided by PIMS (the Pacific Institute for Mathematical Sciences), Cybera, and Compute Canada. To access Syzygy, go to `uleth.syzygy.ca` and log in with your ULeth credentials. Below is a video explaining some of the features of our Jupyter hub.*

YouTube: `https://www.youtube.com/watch?v=VUfp7AQdxhk`

Note: if you click the login button and nothing happens, click the back button and try again. Sometimes there's a problem with our single sign-on service.

*The primary type of document you'll encounter on Syzygy is the **Jupyter notebook**. Content in a Juypter notebook is organized into **cells**. Some cells contain text, which can be in either HTML or **Markdown**. Markdown is a simple markup language. It's not as versatile as HTML, but it's easier to use. On Jupyter, markdown supports the LaTeX language for mathematical expressions. Use single dollar signs for inline math:* `$\frac{d}{dx}\sin(x)=\cos(x)$` *produces $\frac{d}{dx}\sin(x) = \cos(x)$, for example.*

If you want "display math", use double dollar signs. Unfortunately, entering matrices is a bit tedious. For example, `$$A = \begin{bmatrix}1 & 2 & 3\\4 & 5 & 6 &\end{bmatrix}$$` *produces*

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}.$$

Later we'll see how to enter things like matrices in Python.

It's also possible to use markdown to add emphasis, images, URLs, etc.. For details, see the following Markdown cheatsheet[1], or this quick reference[2] from `callysto.ca`.

---

[1]`github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet`
[2]`callysto.ca/wp-content/uploads/2018/12/Callysto-Cheatsheet_12.19.18_web.pdf`

What's cool about a Jupyter notebook is that in addition to markdown cells, which can present content and provide explanation, we can also include code cells. Jupyter supports many different programming languages, but we will stick mainly to Python.

A.2 Python basics

OK, so you've logged into Syzygy and you're ready to write some code. What does basic code look like in Python? The good news is that you don't need to be a programmer to do linear algebra in Python. Python includes many different libraries that keep most of the code under the hood, so all you have to remember is what command you need to use to accomplish a task. That said, it won't hurt to learn a little bit of basic coding.

Basic arithmetic operations are understood, and you can simply type them in. Hit `shift+enter` *in a code cell to execute the code and see the result.*

```
3+4
```

```
3*4
```

```
3**4
```

OK, great. But sometimes we want to do calculations with more than one step. For that, we can assign variables.

```
a = 14
b = -9
c = a+b
print(a, b, c)
```

Sometimes you might need input that's a string, rather than a number. We can do that, too.

```
string_var = "Hey, look at my string!"
print(string_var)
```

Another basic construction is a list. Getting the hang of lists is useful, because in a sense, matrices are just really fancy lists.

```
empty_list = list()
this_too = []
list_of_zeros = [0]*7
print(list_of_zeros)
```

Once you have an empty list, you might want to add something to it. This can be done with the append *command.*

```
empty_list.append(3)
print(empty_list)
print(len(empty_list))
```

Go back and re-run the above code cell two or three more times. What happens? Probably you can guess what the len *command is for. Now let's get really carried away and do some "for real" coding, like loops!*

```
for i in range(10):
    empty_list.append(i)
print(empty_list)
```

Notice the indentation in the second line. This is how Python handles things like for loops, with indentation rather than bracketing. We could say more about lists but perhaps it's time to talk about matrices. For further reading, you can *start here*[1].

A.3 SymPy for linear algebra

SymPy is a Python library for symbolic algebra. On its own, it's not as powerful as programs like Maple, but it handles a lot of basic manipulations in a fairly simple fashion, and when we need more power, it can interface with other Python libraries.

Another advantage of SymPy is sophisticated "pretty-printing". In fact, we can enable MathJax within SymPy, so that output is rendered in the same way as when LaTeX is entered in a markdown cell.

A.3.1 SymPy basics

Running the following Sage cell will load the SymPy library and turn on MathJax.

```
from sympy import *
init_printing()
```

Note: *the command* `from sympy import *` *given above is not best practice. It can be convenient when you want to do a quick calculation (for example, on a test), but can have unintended consequences. It is better to only load those parts of a library you are going to use; for example,* `from sympy import Matrix, init_printing`.

If you are going to be working with multiple libraries, and more than one of them defines a certain command, you can use prefixes to indicate what library you want to use. For example, if you enter `import sympy as sy`, each SymPy command will need to be appended with `sy`; for example, you might write `sy.Matrix` instead of simply `Matrix`. Let's use SymPy to create a 2×3 matrix.

```
from sympy import Matrix, init_printing
init_printing()
A = Matrix(2,3,[1,2,3,4,5,6])
A
```

The `A` on the second line asks Python to print the matrix using SymPy's printing support. If we use Python's `print` command, we get something different; note that the next Sage cell remembers the values from the previous one, if you are using the HTML version of the book.

```
print(A)
```

We'll have more on matrices in Subsection **??**. For now, let's look at some more basic constructions. One basic thing to be mindful of is the type of numbers we're working with. For example, if we enter $2/7$ in a code cell, Python will interpret this as a floating point number (essentially, a division).

---

[1]`developers.google.com/edu/python/lists`

(If you are using Sage cells in HTML rather than Jupyter, this will automatically be interpreted as a fraction.)

```
2/7
```

But we often do linear algebra over the rational numbers, and so SymPy will let you specify this. First, you'll need to load the `Rational` function.

```
from sympy import Rational
Rational(2,7)
```

You might not think to add the comma above, because you're used to writing fractions like $2/7$. Fortunately, the SymPy authors thought of that:

```
Rational(2/7)
```

Hmm... You might have got the output you expected in the cell above, but maybe not. If you got a much worse looking fraction, read on.

Another cool command is the `sympify` command, which can be called with the shortcut `S`. The input 2 is interpreted as an `int` by Python, but `S(2)` is a "SymPy Integer":

```
from sympy import S
S(2)/7
```

Of course, sometimes you do want to use floating point, and you can specify this, too:

```
2.5
```

```
from sympy import Float
Float(2.5)
```

```
Float(2.5e10)
```

One note of caution: `Float` is part of SymPy, and not the same as the core Python `float` command. You can also put decimals into the Rational command and get the corresponding fraction:

```
Rational(0.75)
```

The only thing to beware of is that computers convert from decimal to binary and then back again, and sometimes weird things can happen:

```
Rational(0.2)
```

Of course, there are workarounds. One way is to enter 0.2 as a string:

```
Rational('0.2')
```

Another is to limit the size of the denominator:

```
Rational(0.2).limit_denominator(10**12)
```

Try some other examples above. Some inputs to try are `1.23` *and* `23e-10`

We can also deal with repeating decimals. These are entered as strings, with square brackets around the repeating part. Then we can "sympify":

```
S('0.[1]')
```

Finally, SymPy knows about mathematical constants like e, π, i, which you'll need from time to time in linear algebra. If you ever need ∞, this is entered as `oo`*.*

```
I*I
```

```
I-sqrt(-1)
```

```
from sympy import pi
pi.is_irrational
```

Finally, from time to time you may need to include parameters (variables) in an expression. Typical code for this is of the form `a, b, c = symbols('a b c', real = True, constant = True)`*. Here, we introduce the symbols* `a,b,c` *with the specification that they represent real-valued constants.*

A.3.2 Matrices in SymPy

Here we collect some of the SymPy commands used throughout this text, for ease of reference. For further details, please consult the online documentation[1].

To create a 2×3 matrix, we can write either `A=Matrix(2,3,[1,2,3,4,5,6])` *or* `A=Matrix([[1,2,3],[4,5,6]])`*, where of course the size and entries can be changed to whatever you want. The former method is a bit faster, but once your matrices get a bit bigger, the latter method is less prone to typos.*

```
A=Matrix(2,3,[1,2,3,4,5,6])
B=Matrix([[1,2,3],[4,5,6]])
A,B
```

Also of note: a column vector $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ can be entered using `Matrix([1,2,3])`*. There are also certain built in special matrices. To get an $n \times n$ identity matrix, use* `eye(n)`*. To get an $m \times n$ zero matrix, use* `zeros(m,n)`*, or* `zeros(n)` *for a square matrix. There is also syntax for diagonal matrices, such as* `diag(1,2,3)`*. What's cool is that you can even use this for block diagonal matrices:*

```
A=Matrix(2,2,[1,2,3,4])
B=Matrix(2,2,[5,6,7,8])
D=diag(A,B)
D
```

To get the reduced row-echelon form of the matrix A, simply use `A.rref()`*. Addition, subtraction, and multiplication use the obvious syntax:* `A+B`*,* `A*B`*, etc.. The determinant of a square matrix is given by* `A.det()`*. Inverses can be computed using* `A.inv()` *or* `A**-1`*. The latter is rather natural, since powers in general are entered as* `A**n` *for A^n.*

---

[1] `docs.sympy.org/latest/modules/matrices/matrices.html`

In most cases where you want to reduce a matrix, you're going to want to simply use the rref *function. But there are times where this can be overzealous; for example, if you have a matrix with one or more symbols. One option is to replace* A.rref() *with* A.echelon_form(). *The* echelon_form *function creates zeros in the pivot columns, but does not create leading on ones.*

For example, let's take the matrix $A = \begin{bmatrix} a & 2 & b \\ 2 & 1 & a \\ 2a & b & 3 \end{bmatrix}$. *Note the difference in output between* rref *and* echelon_form.

```
from sympy import Symbol
a = Symbol('a')
b = Symbol('b')
A = Matrix(3,3,[a,2,b,2,1,a,2*a,b,3])
A, A.rref(), A.echelon_form()
```

It is possible to manually perform row operations when you need additional control. This is achieved using the function A.elementary_row_op(<arguments>), *with arguments* op,row,k,row1,row2.

We have the following general syntax:

- *To swap two rows:*

  ○ op='n<->m'

  ○ row1=i, *where* i *is the index of the first row being swapped (remembering that rows are indexed starting with* 0 *for the first row).*

  ○ row2=j, *where* j *is the index of the second row being swapped.*

- *To rescale a row:*

  ○ op='n->kn'

  ○ row=i, *where* i *is the index of the row being rescaled.*

  ○ k=c, *where* c *is the value of the scalar you want to multiply by.*

- *To add a multiple of one row to another:*

  ○ op='n->n+km'

  ○ row=i, *where* i *is the index of the row you want to change.*

  ○ k=c, *where* c *is the multiple of the other row.*

  ○ row2=j, *where* j *is the index of the other row.*

When studying matrix transformations, we are often interested in the null space and column space, since these correspond to the kernel and image of a linear transformation. This is achieved, simply enough, using A.nullspace() *and* A.colspace(). *The output will be a basis of column vectors for these spaces, and these are exactly the ones you'd find doing Gaussian elimination by hand.*

Once you get to orthogonality, you'll want to be able to compute things like dot products, and transpose. These are simple enough. The dot product of vectors X,Y *is simply* X.dot(Y). *The transpose of a matrix* A *is* A.T. *As we should expect,* X\dotp Y = X^TY.

```
X=Matrix(3,1,[1,2,3])
Y=Matrix(3,1,[4,5,6])
X.dot(Y),(X.T)*Y
```

Of course, nobody wants to do things like the Gram Schmidt algorithm by hand. Fortunately, there's a function for that. If we have vectors X,Y,Z, *we can make a list* L=[X,Y,Z], *and perform Gram Schmidt with* GramSchmidt(L). *If you want your output to be an orthonormal basis (and not merely orthogonal), then you can use* GramSchmidt(L,true).

It's useful to note that the output from functions like nullspace() *are automatically treated as lists. So one can use simple code like the following:*

```
from sympy import GramSchmidt
A=Matrix(2,3,[1,0,3,2,-1,4])
L=A.nullspace()
GramSchmidt(L)
```

If for some reason you need to reference particular vectors in a list, this can be done by specifying the index. If L=[X,Y,Z], *then* L[0]==X, L[1]==Y, *and* L[2]==Z.

Next up is eigenvalues and eigenvectors. Given an $n \times n$ matrix A, we have the following:

- *For the characteristic polynomial, use* A.charpoly(). *However, the result will give you something SymPy calls a "PurePoly", and the* factor *command will have no effect. Instead, use* A.charpoly().as_expr().

- *If we know that 3 is an eigenvalue of a 4×4 matrix A, one way to get a basis for the eigenspace $E_3(A)$ is to do:*

$$B=A-3*eye(4)$$
$$B.nullspace()$$

  If you just want all the eigenvalues and eigenvectors without going through the steps, then you can simply execute A.eigenvects(). *The result is a list of lists — each list in the list is of the form: eigenvalue, multiplicity, basis for the eigenspace.*

  For diagonalization, one can do A.diagonalize(). *But this will not necessarily produce orthogonal diagonalization for a symmetric matrix.*

For complex vectors and matrices, the main additional operation we need is the **hermitian conjugate**. *The hermitian conjugate of a matrix* A *is called using* A.H, *which is simple enough. Unfortunately, there is no built-in complex inner product, perhaps because mathematicians and physicists cannot agree on which of the two vectors in the inner product should have the complex conjugate applied to it. Since we define the complex inner product by $\langle \mathbf{z}, \mathbf{w} \rangle = \mathbf{z} \cdot \bar{\mathbf{w}}$, we can execute the inner product in SymPy using* Z.dot(W.H), *or* (W.H)*Z, *although the latter gives the output as a 1×1 matrix rather than a number.*

Don't forget that when entering complex matrices, the complex unit is entered as I. *Also, complex expressions are not simplified by default, so you will often need to wrap your output line in* simplify(). *The Sage Cell below contains complete code for the unitary diagonalization of a 2×2 hermitian matrix with distinct eigenvalues. When doing a problem like this in a Sage cell, it's a good idea to execute each line of code (and display output) before moving on to the next. In this case, printing the output for the list* L *given by* A.eigenvects() *helps explain the complicated-looking definitions of the vectors* v,w. *Of course, if we had a matrix with repeated eigenvalues, we'd need to add steps involving Gram Schmidt.*

```
from sympy import Matrix,init_printing,simplify
init_printing()
```

```
A = Matrix(2,2,[4,3-I,3+I,1])
L = A.eigenvects()
v = ((L[0])[2])[0]
w = ((L[1])[2])[0]
u1 = (1/v.norm())*v
u2 = (1/w.norm())*w
U = u1.row_join(u2)
u1, u2, U, simplify(U.H*A*U)
```

There are a few other commands that might come in handy as you work through this material:

- *Two matrices can be glued together. If matrices* A,B *have the same number of rows, the command* A.row_join(B) *will glue the matrices together, left-to-right. If they have the same number of columns,* A.col_join(B) *will glue them together top-to-bottom.*

- *To insert a column* C *into a matrix* M *(of appropriate size) as the* jth *column, you can do* M.col_insert(j,C). *Just remember that columns are indexed starting at zero, so you might want* j-1 *instead of* j. *This can be useful for things like solving a system* $A\mathbf{x} = B$, *where you want to append the column* B *to the matrix* A.

- *A* QR*-factorization can be performed using* Q,R=A.QRdecomposition()

- *The Jordan canonical form* M *of a matrix* A *can be obtained (along with the matrix* P *whose columns are a Jordan basis) using* P,M=A.jordan_form().

Appendix B

Solutions to Selected Exercises

1 · Vector spaces
1.1 · Abstract vector spaces

Exercise 1.1.4

Solution.

1. Suppose $u + v = u + w$. By adding $-u$ on the left of each side, we obtain:

$$-u + (u + v) = -u + (u + w)$$
$$(-u + u) + v = (-u + u) + w \quad \text{by A3}$$
$$0 + v = 0 + w \quad \text{by A5}$$
$$v = w \quad \text{by A4.}$$

2. We have $c0 = c(0 + 0) = c0 + c0$, by A4 and S2, respectively. Adding $-c0$ to both sides (and using axioms A3, A4, A5 as in the proof of Theorem **??**) we get $0 = c0$.

3. The main difficulty with this problem is getting the logic of the statement correct, and making sure you know what it is you're trying to prove. The desired conclusion is an or statement, which means it suffices to establish one part or the other. Typically, the way to proceed in these cases is to argue that if the first part is false, then the second must be true. This is how we proceed here.

   Suppose that $cv = 0$. If $c = 0$, then it's true that $c = 0$ or $v = 0$, so there's nothing to prove. Suppose then that $c \neq 0$. Then there exists a real number $\frac{1}{c}$ such that $c\left(\frac{1}{c}\right) = 1$. Multiplying both sides of $cv = 0$ by $\frac{1}{c}$, we get

   $$\frac{1}{c}(cv) = \frac{1}{c}0.$$

   By the previous problem, we know that $\frac{1}{c}0 = 0$, and by axioms S4 and S5, we have
   $$\frac{1}{c}(cv) = \left(\frac{1}{c} \cdot c\right)v = 1v = v,$$

   showing that $v = 0$, and thus $c = 0$ or $v = 0$. (This is a proof by cases, relying on the tautology $c = 0$ or $c \neq 0$.)

4. Suppose there are two vectors $0_1, 0_2$ that act as additive identities. Then

   $$0_1 = 0_1 + 0_2 \quad \text{since } v + 0_2 = v \text{ for any } v$$

$$= \mathbf{0}_2 + \mathbf{0}_1 \quad \text{by axiom A2}$$
$$\mathbf{0}_2 \quad \text{since } \mathbf{v} + \mathbf{0}_1 = \mathbf{v} \text{ for any } \mathbf{v}$$

So any two vectors satisfying the property in A4 must, in fact, be the same.

5. *Let* $\mathbf{v} \in V$, *and suppose there are vectors* $\mathbf{w}_1, \mathbf{w}_2 \in V$ *such that* $\mathbf{v} + \mathbf{w}_1 = \mathbf{0}$ *and* $\mathbf{v} + \mathbf{w}_2 = \mathbf{0}$. *Then*

$$\mathbf{w}_1 = \mathbf{w}_1 + \mathbf{0} \quad \text{by A4}$$
$$= \mathbf{w}_1 + (\mathbf{v} + \mathbf{w}_2) \quad \text{by assumption}$$
$$= (\mathbf{w}_1 + \mathbf{v}) + \mathbf{w}_2 \quad \text{by A3}$$
$$= (\mathbf{v} + \mathbf{w}_1) + \mathbf{w}_2 \quad \text{by A2}$$
$$= \mathbf{0} + \mathbf{w}_2 \quad \text{by assumption}$$
$$\mathbf{w}_2 \quad \text{by A4.}$$

1.3 · Span

Exercise 1.3.1

Solution. *This is really asking: are there scalars* s, t *such that*

$$s \begin{bmatrix} 1 \\ 1 \end{bmatrix} + t \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} ?$$

And this, in turn, is equivalent to the system

$$s - t = 2$$
$$s + 2t = 3,$$

which is the same as the matrix equation

$$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Solving the system confirms that there is indeed a solution, so the answer to our original question is yes.

To confirm the above example (and see what the solution is), we can use the computer. This first code cell loads the sympy *Python library, and then configures the output to look nice. For details on the code used below, see the Appendix.*

```
from sympy import Matrix, init_printing
init_printing()
A = Matrix(2,3,[1,-1,2,1,2,3])
A.rref()
```

\[\bbm 1\amp 0\amp \frac73\\0\amp 1\amp \frac13\ebm, (0,1)\]

The above code produces the reduced row-echelon form of the augmented matrix for our system. (The tuple $(0, 1)$ *lists the pivot columns — note that Python indexes the columns starting at* 0 *rather than* 1.*) Do you remember how to get the answer from here? Here's another approach.*

```
B = Matrix(2,2,[1,-1,1,2])
B
```

\[\bbm 1\amp -1\\1\amp 2\ebm\]

```
C = Matrix(2,1,[2,3])
X = (B**-1)*C
X
```

\[\bbm \frac 73\\\frac13\ebm\]

Exercise 1.3.2

Solution.   *We seek scalars s, t such that*

$$s(1 + 2x - 2x^2) + t(3 + 5x + 2x^2) = 1 + x + 4x^2.$$

On the left-hand side, we expand and gather terms:

$$(s + 3t) + (2s + 5t)x + (-2s + 2t)x^2 = 1 + x + 4x^2.$$

These two polynomials are equal if and only if we can solve the system

$$s + 3t = 1$$
$$2s + 5t = 1$$
$$-2s + 2t = 4.$$

We can solve this computationally using matrices again.

```
from sympy import Matrix, init_printing
init_printing()
M = Matrix(3,3,[1,3,1,2,5,1,-2,2,4])
M.rref()
```

\[\bbm 1\amp 0\amp 0\\0\amp 1\amp 0\\0\amp 0\amp 1\ebm,
    (0,1,2)\]

So, what's the answer? Is $p(x)$ in the span? Can we determine what polynomials are in the span? Let's consider a general polynomial $q(x) = a + bx + cx^2$. A bit of thought tells us that the coefficients a, b, c should replace the constants $1, 1, 4$ above.

```
from sympy import symbols
a, b, c = symbols('a b c', real = True, constant = True)
N = Matrix(3,3,[1,3,a,2,5,b,-2,2,c])
N
```

\[\bbm 1\amp 3\amp a\\2\amp 5\amp b\\-2\amp 2\amp c\ebm\]

Asking the computer to reduce this matrix to RREF *won't produce the desired result. But we can always specify row operations.*

```
N1 = N.elementary_row_op(op='n->n+km',row=1,k=-2,row2=0)
N1
```

\[\bbm 1\amp 3\amp a\\0\amp -1\amp -2a+b\\-2\amp 2\amp
    c\ebm\]

In the elementary_row_op *function called above, we are asking the computer to change row 1 (the second row) by adding -2 times row 0)the first row). See Section ?? for complete details on this syntax.*

Now we repeat. Here is another cell to work with:Another option is to replace the rref() *function with the* echelon_form() *function, which doesn't simplify quite as far:*

```
a, b, c = symbols('a b c', real = True, constant = True)
N = Matrix(3,3,[1,3,a,2,5,b,-2,2,c])
N.echelon_form()
```

```
\[\bbm 1\amp 3\amp a\\0\amp -1\amp -2a+b\\0\amp 0\amp
    14a-8b-c\ebm\]
```

For a consistent system, we need $c = 14a - 8b$.

1.5 · Linear Independence

Exercise 1.5.2

Solution.   *Again, we set up a matrix and reduce:*

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix(3,3,[1,-1,-1,2,0,4,0,3,9])
A.rref()
```

```
\[\bbm 1\amp 0\amp 2\\0\amp 1\amp 3\\0\amp 0\amp 0\ebm,
    (0,1)\]
```

Notice that this time we don't get a unique solution, so we can conclude that these vectors are not independent. Furthermore, you can probably deduce from the above that we have $2\mathbf{v}_1 + 3\mathbf{v}_2 - \mathbf{v}_3 = \mathbf{0}$. *Now suppose that* $\mathbf{w} \in span\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$. *In how many ways can we write* $\mathbf{w}$ *as a linear combination of these vectors?*

Exercise 1.5.3

Solution.   *In each case, we set up the defining equation for independence, collect terms, and then analyze the resulting system of equations. (If you work with polynomials often enough, you can probably jump straight to the matrix. For now, let's work out the details.)*

Suppose
$$r(x^2 + 1) + s(x + 1) + tx = 0.$$

Then $rx^2 + (s + t)x + (r + s) = 0 = 0x^2 + 0x + 0$, *so*

$$r = 0$$
$$s + t = 0$$
$$r + s = 0.$$

And in this case, we don't even need to ask the computer. The first equation gives $r = 0$ *right away, and putting that into the third equation gives* $s = 0$, *and the second equation then gives* $t = 0$.

Since $r = s = t = 0$ *is the only solution, the set is independent.*

Repeating for S_2 *leads to the equation*

$$(r + 2s + t)x^2 + (-r + s + 5t)x + (3r + 5s + t)1 = 0.$$

This gives us:

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix(3,3,[1,2,1,-1,1,5,3,5,1])
A.rref()
```

\[\bbm 1\amp 0\amp -3\\0\amp 1\amp 2\\0\amp 0\amp 0\ebm,
    (0,1)\]

Exercise 1.5.4

Solution.   *Again, we set a linear combination equal to the zero vector, and combine:*

$$a\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} + b\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + c\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + d\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -a+b+c & -b+c-d \\ -b+c-d & -a+b+c \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

We could proceed, but we might instead notice right away that equations 1 and 4 are identical, and so are equations 2 and 3. With only two distinct equations and 4 unknowns, we're certain to find nontrivial solutions.

1.6 · Basis and dimension

Exercise 1.6.5

Solution.   *Let $X = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Then $AX = \begin{bmatrix} a+c & b+d \\ 0 & 0 \end{bmatrix}$, and $XA = \begin{bmatrix} a & a \\ c & c \end{bmatrix}$, so the condition $AX = XA$ requires:*

$$a + c = a$$
$$b + d = a$$
$$0 = c$$
$$0 = c.$$

So $c = 0$, in which case the first equation $a = a$ is trivial, and we are left with the single equation $a = b + d$. Thus, our matrix X must be of the form

$$X = \begin{bmatrix} b+d & b \\ 0 & d \end{bmatrix} = b\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} + d\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since the matrices $\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ are not scalar multiples, they must be independent, and therefore, they form a basis for U.

Exercise 1.6.7

(a) **Solution.**   *We need to show that the set is independent, and that it spans.*

The set is independent if the equation

$$x(1, 1, 0) + y(1, 0, 1) + z(0, 1, 1) = (0, 0, 0)$$

has $x = y = z = 0$ as its only solution. This equation is equivalent to the system

$$x + y = 0$$
$$x + z = 0$$
$$y + z = 0.$$

We know that the solution to this system is unique if the coefficient matrix $A =$
$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ is invertible. Note that the columns of this matrix are vectors in our
set.

We can determine invertibility either by showing that the RREF of A is the identity, or by showing that the determinant of A is nonzero. Either way, this is most easily done by the computer:

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix([[1,1,0],[1,0,1],[0,1,1]])
A.rref(), A.det()
```

\[\left(\bbm 1\amp 0\amp 0\\0\amp 1\amp 0\\0\amp
    1\ebm, (0,1,2)\right), -2\]

Our set of vectors is therefore linearly independent. Now, to show that it spans, we need to show that for any vector (a, b, c), the equation

$$x(1, 1, 0) + y(1, 0, 1) + z(0, 1, 1) = (a, b, c)$$

has a solution. But we know that this system has the same coefficient matrix as the one above, and that existence of a solution again follows from invertibility of A, which we have already established.

Note that for three vectors in $\mathbb{R}^3$, once independence has been confirmed, span is automatic. We will soon see that this is not a coincidence.

(b) **Solution.**   Based on what we learned from the first set, determining whether or not this set is a basis is equivalent to determining whether or not the matrix A whose columns consist of the vectors in the set is invertible. We form the matrix
$$A = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$
and then check invertibility using the computer.

```
A = Matrix([[-1,1,1],[1,-1,1],[1,1,-1]])
A.det()
```

\[4\]

Since the determinant is nonzero, our set is a basis.

Exercise 1.6.10

Solution.

a By definition, $U_1 = \text{span}\{1 + x, x + x^2\}$, and these vectors are independent, since neither is a scalar multiple of the other. Since there are two vectors in this basis, $\dim U_1 = 2$.

b If $p(1) = 0$, then $p(x) = (x - 1)q(x)$ for some polynomial q. Since U_2 is a subspace of P_2, the degree of q is at most 2. Therefore, $q(x) = ax + b$ for some $a, b \in \mathbb{R}$, and

$$p(x) = (x - 1)(ax + b) = a(x^2 - x) + b(x - 1).$$

Since p was arbitrary, this shows that $U_2 = \text{span}\{x^2 - x, x - 1\}$.

The set $\{x^2 - x, x - 1\}$ is also independent, since neither vector is a scalar multiple of the other. Therefore, this set is a basis, and $\dim U_2 = 2$.

c If $p(x) = p(-x)$, then $p(x)$ is an even polynomial, and therefore $p(x) = a + bx^2$ for $a, b \in \mathbb{R}$. (If you didn't know this it's easily verified: if

$$a + bx + cx^2 = a + b(-x) + c(-x)^2,$$

we can immediately cancel a from each side, and since $(-x)^2 = x^2$, we can cancel cx^2 as well. This leaves $bx = -bx$, or $2bx = 0$, which implies that $b = 0$.)

It follows that the set $\{1, x^2\}$ spans U_3, and since this is a subset of the standard basis $\{1, x, x^2\}$ of P_2, it must be independent, and is therefore a basis of U_3, letting us conclude that $\dim U_3 = 2$.

Exercise 1.6.14

Solution.   By the previous theorem, we can form a basis by adding vectors from the standard basis

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right\}.$$

It's easy to check that $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ is not in the span of $\{\mathbf{v}, \mathbf{w}\}$. To get a basis, we need one more vector. Observe that all three of our vectors so far have a zero in the $(2,1)$-entry. Thus, $\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ cannot be in the span of the first three vectors, and adding it gives us our basis.

Exercise 1.6.15

Solution.   Again, we only need to add one vector from the standard basis $\{1, x, x^2, x^3\}$, and it's not too hard to check that any of them will do.

2 · Linear Transformations
2.1 · Definition and examples

Exercise 2.1.6

Solution.   Since we know the value of T on the standard basis, we can use properties of linear transformations to immediately obtain the answer:

$$\begin{aligned}
T \begin{bmatrix} -2 \\ 4 \end{bmatrix} &= T \left(-2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\
&= -2T \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4T \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
&= -2 \begin{bmatrix} 3 \\ -4 \end{bmatrix} + 4 \begin{bmatrix} 5 \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} 14 \\ 16 \end{bmatrix}.
\end{aligned}$$

Exercise 2.1.7

Solution.   At first, this example looks the same as the one above, and to some extent, it is. The difference is that this time, we're given the values of T on a basis that is not the standard one. This means we first have to do some work to determine how to write the given vector in terms of the given basis.

Suppose we have $a \begin{bmatrix} 3 \\ 1 \end{bmatrix} + b \begin{bmatrix} 2 \\ -5 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ for scalars a, b. This is equivalent to the

matrix equation

$$\begin{bmatrix} 3 & 2 \\ 1 & -5 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}.$$

Solving (perhaps using the code cell below), we get $a = \frac{26}{17}, b = -\frac{5}{17}$.

```
from sympy import Matrix,init_printing
init_printing()
A = Matrix(2,2,[3,2,1,-5])
B = Matrix(2,1,[4,3])
(A**-1)*B
```

\[\bbm \frac{26}{17}\\-\frac{5}{17}\ebm\]

Therefore,

$$T\begin{bmatrix} 4 \\ 3 \end{bmatrix} = \frac{26}{17}\begin{bmatrix} 1 \\ 4 \end{bmatrix} - \frac{5}{17}\begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 16/17 \\ 109/17 \end{bmatrix}.$$

Exercise 2.1.8

Solution. *We need to find scalars a, b, c such that*

$$2 - x + 3x^2 = a(x + 2) + b(1) + c(x^2 + x).$$

We could set up a system and solve, but this time it's easy enough to just work our way through. We must have $c = 3$, to get the correct coefficient for x^2. This gives

$$2 - x + 3x^2 = a(x + 2) + b(1) + 3x^2 + 3x.$$

Now, we have to have $3x + ax = -x$, so $a = -4$. Putting this in, we get

$$2 - x + 3x^2 = -4x - 8 + b + 3x^2 + 3x.$$

Simiplifying this leaves us with $b = 10$. Finally, we find:

$$\begin{aligned} T(2 - x + 3x^2) &= T(-4(x + 2) + 10(1) + 3(x^2 + x)) \\ &= -4T(x + 2) + 10T(1) + 3T(x^2 + x) \\ &= -4(1) + 10(5) + 3(0) = 46. \end{aligned}$$

Exercise 2.1.9

Solution. *Since $\{(1, 2), (-1, 1)\}$ forms a basis of $\mathbb{R}^2$ (the vectors are not parallel and there are two of them), it suffices to determine how to write a general vector in terms of this basis. Suppose*

$$x(1, 2) + y(-1, 1) = (a, b)$$

for a general element $(a, b) \in \mathbb{R}^2$. This is equivalent to the matrix equation $\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$, which we can solve as $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} a \\ b \end{bmatrix}$:

```
from sympy import Matrix, init_printing, symbols
init_printing()
a, b = symbols('a b', real = True, constant = True)
A = Matrix(2,2,[1,-1,2,1])
B = Matrix(2,1,[a,b])
(A**-1)*B
```

```
\[\bbm
    \frac{a}{3}+\frac{b}{3}\\-\frac{2a}{3}+\frac{b}{3}\ebm\]
```

This gives us the result

$$(a, b) = \frac{1}{3}(a + b) \cdot (1, 2) + \frac{1}{3}(-2a + b) \cdot (-1, 1).$$

Thus,

$$T(a, b) = \frac{1}{3}(a + b) \cdot T(1, 2) + \frac{1}{3}(-2a + b) \cdot T(-1, 1)$$
$$= \frac{1}{3}(a + b) \cdot (1, 1, 0) + \frac{1}{3}(-2a + b) \cdot (0, 2, -1)$$
$$= \left(\frac{a + b}{3}, -a + b, \frac{2a - b}{3} \right).$$

We conclude that

$$T(3, 2) = \left(\frac{5}{3}, -1, \frac{4}{3} \right).$$

Exercise 2.1.10

Solution.   *Let us suppose that $\{T(\mathbf{v}_1), \ldots, T(\mathbf{v}_n)\}$ is linearly independent. We want to show that the set $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ is linearly independent. To that end, suppose that we have*

$$c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n = \mathbf{0}$$

for some scalars $c_1, \ldots, c_n$ (that we want to show must all equal zero).

We want to make use of our hypothesis, so we need to bring the linear map T into the picture. We apply T to both sides of the equation above, giving us:

$$T(c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n) = T(\mathbf{0}).$$

Now we make use of both parts of Theorem ?? to get

$$c_1T(\mathbf{v}_1) + \cdots + c_nT(\mathbf{v}_n) = \mathbf{0}.$$

By our hypothesis that the $T(\mathbf{v}_i)$ are independent, we must conclude that all the scalars are zero, and we're done.

2.2 · Kernel and Image

Exercise 2.2.8

Solution.

1. *Suppose $T : V \to W$ is injective. Then $\ker T = \{0\}$, so*

$$\dim V = 0 + \dim \operatorname{im} T \leq \dim W,$$

since $\operatorname{im} T$ is a subspace of W.

Conversely, suppose $\dim V \leq \dim W$. Choose a basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ of V, and a basis $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ of W, where $m \leq n$. By Theorem ??, there exists a linear transformation $T : V \to W$ with $T(\mathbf{v}_i) = \mathbf{w}_i$ for $i = 1, \ldots, m$. (The main point here is that we run out of basis vectors for V before we run out of basis vectors for W.) This map is injective: if $T(\mathbf{v}) = \mathbf{0}$, write $\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_m\mathbf{v}_m$. Then

$$\mathbf{0} = T(\mathbf{v})$$

$$= T(c_1\mathbf{v}_1 + \cdots + c_m\mathbf{v}_m)$$
$$= c_1T(\mathbf{v}_1) + \cdots + c_mT(\mathbf{v}_m)$$
$$= c_1\mathbf{w}_1 + \cdots + c_m\mathbf{w}_m.$$

Since $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ is a subset of a basis, it's independent. Therefore, the scalars c_i must all be zero, and therefore $\mathbf{v} = \mathbf{0}$.

2. Suppose $T : V \to W$ is surjective. Then $\dim \operatorname{im} T = \dim W$, so

$$\dim V = \dim \ker T + \dim W \geq \dim W.$$

Conversely, suppose $\dim V \geq \dim W$. Again, choose a basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ of V, and a basis $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ of W, where this time, $m \geq n$. We can define a linear transformation as follows:

$$T(\mathbf{v}_1) = \mathbf{w}_1, \ldots, T(\mathbf{v}_n) = \mathbf{w}_n, \text{ and } T(\mathbf{v}_j) = \mathbf{0} \text{ for } j > n.$$

It's easy to check that this map is a surjection: given $\mathbf{w} \in W$, we can write it in terms of our basis as $\mathbf{w} = c_1\mathbf{w}_1 + \cdots + c_n\mathbf{w}_n$. Using these same scalars, we can define $\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n \in V$ such that $T(\mathbf{v}) = \mathbf{w}$.

Note that it's not important how we define $T(\mathbf{v}_j)$ when $j > n$. The point is that this time, we run out of basis vectors for W before we run out of basis vectors for V. Once each vector in the basis of W is in the image of T, we're guaranteed that T is surjective, and we can define the value of T on any remaining basis vectors however we want.

2.3 · Isomorphisms (a.k.a. invertible linear maps)

Exercise 2.3.6

Solution. Suppose we have linear maps $U \xrightarrow{T} V \xrightarrow{S} W$, and let $\mathbf{u}_1, \mathbf{u}_2 \in U$. Then

$$ST(\mathbf{u}_1 + \mathbf{u}_2) = S(T(\mathbf{u}_1 + \mathbf{u}_2))$$
$$= S(T(\mathbf{u}_1) + T(\mathbf{u}_2))$$
$$= S(T(\mathbf{u}_1)) + S(T(\mathbf{u}_2))$$
$$= ST(\mathbf{u}_1) + ST(\mathbf{u}_2),$$

and for any scalar c,

$$ST(c\mathbf{u}_1) = S(T(c\mathbf{u}_1)) = S(cT(\mathbf{u}_1)) = cS(T(\mathbf{u}_1)) = c(ST(\mathbf{u}_1)).$$

Exercise 2.3.7

Solution. Let $\mathbf{w}_1, \mathbf{w}_2 \in W$. Then there exist $\mathbf{v}_1, \mathbf{v}_2 \in V$ with $\mathbf{w}_1 = T(\mathbf{v}_1), \mathbf{w}_2 = T(\mathbf{v}_2)$. We then have

$$T^{-1}(\mathbf{w}_1 + \mathbf{w}_2) = T^{-1}(T(\mathbf{v}_1) + T(\mathbf{v}_2))$$
$$= T^{-1}(T(\mathbf{v}_1 + \mathbf{v}_2))$$
$$= \mathbf{v}_1 + \mathbf{v}_2$$
$$= T^{-1}(\mathbf{w}_1) + T^{-1}(\mathbf{w}_2).$$

For any scalar c, we similarly have

$$T^{-1}(c\mathbf{w}_1) = T^{-1}(cT(\mathbf{v}_1)) = T^{-1}(T(c\mathbf{v}_1)) = c\mathbf{v}_1 = cT^{-1}(\mathbf{w}_1).$$

Exercise 2.3.8

Hint. This is really a Math 2000 problem.

3 · Orthogonality and Applications
3.1 · Orthogonal sets of vectors
3.1.1 · Basic definitions and properties

Exercise 3.1.4

Solution. Note that the distributive property, together with symmetry, let us handle this dot product using what is essentially "FOIL":

$$(4\boldsymbol{x} - 3\boldsymbol{y})\cdot(\boldsymbol{x} + 5\boldsymbol{y}) = (4\boldsymbol{x})\cdot\boldsymbol{x} + (4\boldsymbol{x})\cdot(5\boldsymbol{y}) + (-3\boldsymbol{y})\cdot\boldsymbol{x} + (-3\boldsymbol{y})\cdot(5\boldsymbol{y})$$
$$= 4(\boldsymbol{x}\cdot\boldsymbol{x}) + (4 \cdot 5)(\boldsymbol{x}\cdot\boldsymbol{y}) - 3(\boldsymbol{y}\cdot\boldsymbol{x}) + (-3 \cdot 5)(\boldsymbol{y}\cdot\boldsymbol{y})$$
$$= 4\|\boldsymbol{x}\|^2 + 20\boldsymbol{x}\cdot\boldsymbol{y} - 3\boldsymbol{x}\cdot\boldsymbol{y} - 15\|\boldsymbol{y}\|^2$$
$$= 4(9) + 17(-2) - 15(1) = -13.$$

Exercise 3.1.5

Solution. This is simply an exercise in properties of the dot product. We have

$$\|\boldsymbol{x} + \boldsymbol{y}\|^2 = (\boldsymbol{x} + \boldsymbol{y})\cdot(\boldsymbol{x} + \boldsymbol{y})$$
$$= \boldsymbol{x}\cdot\boldsymbol{x} + \boldsymbol{x}\cdot\boldsymbol{y} + \boldsymbol{y}\cdot\boldsymbol{x} + \boldsymbol{y}\cdot\boldsymbol{y}$$
$$= \|\boldsymbol{x}\|^2 + 2\boldsymbol{x}\cdot\boldsymbol{y} + \|\boldsymbol{y}\|^2.$$

Exercise 3.1.6

Solution. If $\boldsymbol{x} = \boldsymbol{0}$, then the result follows immediately from the dot product formula in Definition ??. Conversely, suppose $\boldsymbol{x}\cdot\boldsymbol{v}_i = 0$ for each i. Since the $\boldsymbol{v}_i$ span $\mathbb{R}^n$, there must exist scalars $c_1, c_2, \ldots, c_k$ such that $\boldsymbol{x} = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_k\boldsymbol{v}_k$. But then

$$\boldsymbol{x}\cdot\boldsymbol{x} = \boldsymbol{x}\cdot(c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_k\boldsymbol{v}_k)$$
$$= c_1(\boldsymbol{x}\cdot\boldsymbol{v}_1) + c_2(\boldsymbol{x}\cdot\boldsymbol{v}_2) + \cdots + c_k(\boldsymbol{x}\cdot\boldsymbol{v}_k)$$
$$= c_1(0) + c_2(0) + \cdots + c_k(0) = 0.$$

3.1.2 · Orthogonal sets of vectors

Exercise 3.1.10

Solution. Clearly, all three vectors are nonzero. To confirm the set is orthogonal, we simply compute dot products:

$$(1, 0, 1, 0)\cdot(-1, 0, 1, 1) = -1 + 0 + 1 + 0 = 0$$
$$(-1, 0, 1, 1)\cdot(1, 1, -1, 2) = -1 + 0 - 1 + 2 = 0$$
$$(1, 0, 1, 0)\cdot(1, 1, -1, 2) = 1 + 0 - 1 + 0 = 0.$$

To find a fourth vector, we proceed as follows. Let $\boldsymbol{x} = (a, b, c, d)$. We want $\boldsymbol{x}$ to be orthogonal to the three vectors in our set. Computing dot products, we must have:

$$(a, b, c, d)\cdot(1, 0, 1, 0) = a + c = 0$$
$$(a, b, c, d)\cdot(-1, 0, 1, 1) = -a + c + d = 0$$
$$(a, b, c, d)\cdot(1, 1, -1, 2) = a + b - c + 2d = 0.$$

This is simply a homogeneous system of three equations in four variables. Using the Sage cell below, we find that our vector must satisfy $a = \frac{1}{2}d, b = -3d, c = -\frac{1}{2}d$.

```
from sympy import Matrix, init_printing
init_printing()
A=Matrix(3,4,[1,0,1,0,-1,0,1,1,1,1,-1,2])
A.rref()
```

```
\[\bbm 1\amp 0\amp 0\amp -\frac12\\ 0\amp 1\amp 0\amp 3\\
    0\amp 0\amp 1\amp \frac12\ebm, (0,1,2)\]
```

One possible nonzero solution is to take $d = 2$, giving $\boldsymbol{x} = (1, -6, -1, 2)$. *We'll leave the verification that this vector works as an easy exercise.*

Exercise 3.1.14

Solution.   We compute

$$\left(\frac{\boldsymbol{v}\cdot\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|^2}\right)\boldsymbol{x}_1 + \left(\frac{\boldsymbol{v}\cdot\boldsymbol{x}_2}{\|\boldsymbol{x}_2\|^2}\right)\boldsymbol{x}_2 + \left(\frac{\boldsymbol{v}\cdot\boldsymbol{x}_3}{\|\boldsymbol{x}_3\|^2}\right)\boldsymbol{x}_3$$

$$= \frac{4}{2}\boldsymbol{x}_1 + \frac{-9}{3}\boldsymbol{x}_2 + \frac{-28}{7}\boldsymbol{x}_3$$

$$= 2(1,0,1,0) - 3(-1,0,1,1) - 4(1,1,-1,2)$$

$$= (1, -4, 3, -11) = \boldsymbol{v},$$

so $\boldsymbol{v} \in span\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$.

On the other hand, repeating the same calculation with $\boldsymbol{w}$, we find

$$\left(\frac{\boldsymbol{v}\cdot\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|^2}\right)\boldsymbol{x}_1 + \left(\frac{\boldsymbol{v}\cdot\boldsymbol{x}_2}{\|\boldsymbol{x}_2\|^2}\right)\boldsymbol{x}_2 + \left(\frac{\boldsymbol{v}\cdot\boldsymbol{x}_3}{\|\boldsymbol{x}_3\|^2}\right)\boldsymbol{x}_3$$

$$= \frac{1}{2}(1,0,1,0) - \frac{5}{3}(-1,0,1,1) + \frac{4}{7}(1,1,-1,2)$$

$$= \left(\frac{73}{42}, \frac{4}{7}, -\frac{115}{42}, -\frac{11}{21}\right) \neq \boldsymbol{w},$$

so $\boldsymbol{w} \notin span\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$.

Soon, we'll see that the quantity we computed when showing that $\boldsymbol{w} \notin span\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ is, in fact, the orthogonal projection of $\boldsymbol{w}$ onto the subspace $span\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$.

3.2 · Orthogonal Projection
3.2.1 · The Gram-Schmidt Procedure

Exercise 3.2.3

Solution.   First, note that we can actually jump right into the Gram-Schmidt procedure. If the set B is not a basis, then it won't be independent, and when we attempt to construct the third vector in our orthonormal basis, its projection on the the subspace spanned by the first two will be the same as the original vector, and we'll get zero when we subtract the two.

We let $\boldsymbol{x}_1 = (1, -2, 1), \boldsymbol{x}_2 = (3, 0, -2), \boldsymbol{x}_3 = (-1, 1, 2)$, and set $\boldsymbol{v}_1 = \boldsymbol{x}_1$. Then we have

$$\boldsymbol{v}_2 = \boldsymbol{x}_2 - \left(\frac{\boldsymbol{x}_2\cdot\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|^2}\right)\boldsymbol{v}_1$$

$$= (3, 0, -2) - \frac{1}{6}(1, -2, 1)$$

$$= \frac{1}{6}(17, 2, -3).$$

Next, we compute $\boldsymbol{v}_3$.

$$\boldsymbol{v}_3 = \boldsymbol{x}_3 - \left(\frac{\boldsymbol{x}_3\cdot\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|^2}\right)\boldsymbol{v}_1 - \left(\frac{\boldsymbol{x}_3\cdot\boldsymbol{v}_2}{\|\boldsymbol{v}_2\|^2}\right)\boldsymbol{v}_2$$

$$= (-1, 1, 2) - \frac{-1}{6}(1, -2, 1) - \cdot\frac{-21}{303}(17, 2, -3)$$

$$= (-1, 1, 2) + \frac{1}{6}(1, -2, 1) + \frac{7}{101}(17, 2, -3)$$

$$= \frac{1}{606} \left((-606, 606, 1212) + (101, -202, 101) + (782, 84, -126) \right)$$

$$= \frac{1}{606} (277, 488, 1187).$$

OK. Now, given the frequency with which typos occur in this text, and the fact that I tried to do the above problem in my head while typing (with an occasional calculator check), there's a good chance there's a mistake somewhere. Let's check our work.

```
from sympy import Matrix, init_printing, GramSchmidt
init_printing()
L=(Matrix([1,-2,1]),Matrix([3,0,-2]),Matrix([-1,1,2]))
GramSchmidt(L)
```

```
\[\left[\bbm 1\\-2\\1\ebm, \bbm \frac{17}{6}\\ \frac13\\
    -\frac{13}{6}\ebm, \bbm \frac{52}{77}\\ \frac{65}{77}\\
    \frac{78}{77}\ebm\right]\]
```

3.2.2 · Projections

Exercise 3.2.9

Hint.   Suppose we find vectors $\boldsymbol{p}$ and $\boldsymbol{p}'$ using basis B and B'. Note that $\boldsymbol{p} - \boldsymbol{p}' \in U$, but also that

$$\boldsymbol{p} - \boldsymbol{p}' = (\boldsymbol{p} - \boldsymbol{x}) - (\boldsymbol{p}' - \boldsymbol{x})$$

Now use Theorem ??

Exercise 3.2.11

Solution.   First, we note that for a general element of U, we have

$$(a-b+3c, 2a+b, 3c, 4a-b+3c, a-4c) = a(1,2,0,4,1)+b(-1,1,0,-1,0)+c(3,0,3,3,-4).$$

If $\boldsymbol{w} = (x_1, x_2, x_3, x_4, x_5) \in U^\perp$, then we must have

$$\boldsymbol{w} \cdot (1,2,0,4,1) = x_1 + 2x_2 + 4x_4 + x_5 = 0$$
$$\boldsymbol{w} \cdot (-1,1,0,-1,0) = -x_1 + x_2 - x_4 = 0$$
$$\boldsymbol{w} \cdot (3,0,3,3,-4) = 3x_1 + 3x_3 + 3x_4 - 4x_5 = 0.$$

To find a basis for $U^\perp$, we simply need to find the nullspace of the coefficient matrix for this system, which we do below.

```
from sympy import Matrix, init_printing
init_printing()
A = Matrix(3,5,[1,2,0,4,1,-1,1,0,-1,0,3,0,3,3,-4])
A.nullspace()
```

```
\[\left[\bbm -2\\-1\\1\\1\\0\ebm, \bbm -\frac13\\ -\frac13\\
    \frac53\\0\\1\ebm\right]\]
```

4 · Diagonalization
4.2 · Diagonalization of symmetric matrices

Exercise 4.2.1

Solution.   Take $\boldsymbol{x} = \boldsymbol{e}_i$ and $\boldsymbol{y} = \boldsymbol{e}_j$, where $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ is the standard basis for $\mathbb{R}^n$.

Then with $A = [a_{ij}]$ we have

$$a_{ij} = \boldsymbol{e}_i \cdot (A\boldsymbol{e}_j) = (A\boldsymbol{e}_i) \cdot \boldsymbol{e}_j = a_{ji},$$

which shows that $A^T = A$.

Exercise 4.2.7

Solution.   We'll solve this problem with the help of the computer.

```
from sympy import Matrix,init_printing,factor
init_printing()
A = Matrix(3,3,[5,-2,-4,-2,8,-2,-4,-2,5])
p=A.charpoly().as_expr()
factor(p)
```

\[\lambda(\lambda-9)^2\]

We get $c_A(x) = x(x-9)^2$, so our eigenvalues are 0 and 9. For 0 we have $E_0(A) = $ null(A):

```
A.nullspace()
```

\[\bbm 1\\ \frac12\\1\ebm\]

For 9 we have $E_9(A) = $ null$(A - 9I)$.

```
from sympy import eye
B=A-9*eye(3)
B.nullspace()
```

\[\left[\bbm -\frac12\\1\\0\ebm, \bbm -1\\0\\1\ebm\right]\]

The approach above is useful as we're trying to remind ourselves how eigenvalues and eigenvectors are defined and computed. Eventually we might want to be more efficient. Fortunately, there's a command for that.

```
A.eigenvects()
```

\[\left[\left(0,1,\bbm 1\\ \frac12\\1\ebm\right),
    \left(9,2,\left[\bbm -\frac12\\1\\0\ebm,\bbm
    -1\\0\\1\ebm\right]\right)\right]\]

Note that the output above lists each eigenvalue, followed by its multiplicity, and then the associated eigenvectors.

This gives us a basis for $\mathbb{R}^3$ consisting of eigenvalues of A, but we want an orthogonal basis. Note that the eigenvector corresponding to $\lambda = 0$ is orthogonal to both of the eigenvectors corresponding to $\lambda = 9$. But these eigenvectors are not orthogonal to each other. To get an orthogonal basis for $E_9(A)$, we apply the Gram-Schmidt algorithm.

```
from sympy import GramSchmidt
L=B.nullspace()
GramSchmidt(L)
```

```
\[\left[\bbm -\frac12\\1\\0\ebm, \bbm
   -\frac45\\-\frac25\\1\ebm\right]\]
```

This gives us an orthogonal basis of eigenvectors. Scaling to clear fractions, we have

$$\left\{ \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} -4 \\ -2 \\ 5 \end{bmatrix} \right\}$$

From here, we need to normalize each vector to get the matrix P. But we might not like that the last vector has norm $\sqrt{45}$. One option to consider is to apply Gram-Schmidt with the vectors in the other order.

```
L=[Matrix(3,1,[-1,0,1]),Matrix(3,1,[-1,2,0])]
GramSchmidt(L)
```

```
\[\left[\bbm -1\\0\\1\ebm, \bbm
   -\frac12\\2\\-\frac12\ebm\right]\]
```

That gives us the (slightly nicer) basis

$$\left\{ \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -4 \\ 1 \end{bmatrix} \right\}.$$

The corresponding orthonormal basis is

$$B = \left\{ \frac{1}{3} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{18}} \begin{bmatrix} 1 \\ -4 \\ 1 \end{bmatrix} \right\}.$$

This gives us the matrix $P = \begin{bmatrix} 2/3 & -1/\sqrt{2} & 1/\sqrt{18} \\ 1/3 & 0 & -4/\sqrt{18} \\ 2/3 & 1/\sqrt{2} & 1/\sqrt{18} \end{bmatrix}$. Let's confirm that P is orthogonal.

```
P=Matrix(3,3,[2/3, -1/sqrt(2),1/sqrt(18),
   1/3,0,-4/sqrt(18),2/3,1/sqrt(2),1/sqrt(18)])
P,P*P.transpose()
```

```
\[\left(\bbm \frac23 \amp -\frac{\sqrt{2}}{2}\amp
   \frac{\sqrt{2}}{6}\\ \frac13\amp 0\amp
   -\frac{2\sqrt{2}}{3}\\ \frac23 \amp \frac{\sqrt{2}}{2}
   \amp \frac{\sqrt{2}}{6}\ebm,
 \bbm 1\amp 0\amp 0\\0\amp 1\amp 0\\0\amp 0\amp 1\ebm\right)\]
```

Since $PP^T = I_3$, we can conclude that $P^T = P^{-1}$, so P is orthogonal, as required. Finally, we diagonalize A.

```
Q=P.transpose()
Q*A*P
```

```
\[\bbm 0\amp 0\amp 0\\0\amp 9\amp 0\\0\amp 0\amp 9\ebm\]
```

Incidentally, the SymPy library for Python does have a diagaonalization routine; however, it does not do orthogonal diagonalization by default. Here is what it provides for our matrix A.

```
A.diagonalize ()
```

```
\[\left(\bbm 2\amp -1\amp -1\\1\amp 2\amp 0\\2\amp 0\amp
    1\ebm, \bbm 0\amp 0\amp 0\\0\amp 9\amp 0\\0\amp 0\amp
    9\ebm\right)\]
```

4.4 · Diagonalization of complex matrices
4.4.3 · Complex matrices

Exercise 4.4.7

Solution.   We have $\bar{A} = \begin{bmatrix} 4 & 1+i & -2-3i \\ 1-i & 5 & -7i \\ -2+3i & 7i & -4 \end{bmatrix}$, so

$$A^H = (\bar{A})^T = \begin{bmatrix} 4 & 1-i & -2+3i \\ 1+i & 5 & 7i \\ -2-3i & -7i & -4 \end{bmatrix} = A,$$

and

$$BB^H = \frac{1}{4} \begin{bmatrix} 1+i & \sqrt{2} \\ 1-i & \sqrt{2}i \end{bmatrix} \begin{bmatrix} 1-i & 1+i \\ \sqrt{2} & -\sqrt{2}i \end{bmatrix}$$
$$= \frac{1}{4} \begin{bmatrix} (1+i)(1-i)+2 & (1+i)(1+i)-2i \\ (1-i)(1-i)+2i & (1-i)(1+i)+2 \end{bmatrix}$$
$$= \frac{1}{4} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

so that $B^H = B^{-1}$.

Exercise 4.4.11

Solution.   Confirming that $A^H = A$ is almost immediate. We will use the computer below to compute the eigenvalues and eigenvectors of A, but it's useful to attempt this at least once by hand. We have

$$\det(zI - A) = \det \begin{bmatrix} z-4 & -3+i \\ -3-i & z-1 \end{bmatrix}$$
$$(z-4)(z-1) - (-3-i)(-3+i)$$
$$z^2 - 5z + 4 - 10$$
$$(z+1)(z-6),$$

so the eigenvalues are $\lambda_1 = -1$ and $\lambda_2 = 6$, which are both real, as expected.

Finding eigenvectors can seem trickier than with real numbers, mostly because it is no longer immediately apparent when one row or a matrix is a multiple of another. But we know that the rows of $A - \lambda I$ must be parallel for a 2×2 matrix, which lets proceed nonetheless.

For $\lambda_1 = -1$, we have

$$A + I = \begin{bmatrix} 5 & 3-i \\ 3+i & 2 \end{bmatrix}.$$

There are two ways one can proceed from here. We could use row operations to get to the reduced row-echelon form of A. If we take this approach, we multiply row 1 by $\frac{1}{5}$, and then take $-3 - i$ times the new row 1 and add it to row 2, to create a zero, and so on.

Easier is to realize that if we haven't made a mistake calculating our eigenvalues, then the above matrix can't be invertible, so there must be some nonzero vector in the kernel. If $(A + I)\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, then we must have

$$5a + (3 - i)b = 0,$$

when we multiply by the first row of A. This suggests that we take $a = 3 - i$ and $b = -5$, to get $\mathbf{z} = \begin{bmatrix} 3 - i \\ -5 \end{bmatrix}$ as our first eigenvector. To make sure we've done things correctly, we multiply by the second row of $A + I$:

$$(3 + i)(3 - i) + 2(-5) = 10 - 10 = 0.$$

Success! Now we move onto the second eigenvalue.
    For $\lambda_2 = 6$, we get

$$A - 6I = \begin{bmatrix} -2 & 3 - i \\ 3 + i & -5 \end{bmatrix}.$$

If we attempt to read off the answer like last time, the first row of $A - 6I$ suggests the vector $\mathbf{w} = \begin{bmatrix} 3 - i \\ 2 \end{bmatrix}$. Checking the second row to confirm, we find:

$$(3 + i)(3 - i) - 5(2) = 10 - 10 = 0,$$

as before.
    Finally, we note that

$$\langle \mathbf{z}, \mathbf{w} \rangle = (3 - i)\overline{(3 - i)} + (-5)(2) = (3 - i)(3 + i) - 10 = 0,$$

so the two eigenvectors are orthogonal, as expected. We have

$$\|\mathbf{z}\| = \sqrt{10 + 25} = \sqrt{35} \quad \text{and} \quad \|\mathbf{w}\| = \sqrt{10 + 4} = \sqrt{14},$$

so our orthogonal matrix is

$$U = \begin{bmatrix} \frac{3-i}{\sqrt{35}} & \frac{3-i}{\sqrt{14}} \\ -\frac{5}{\sqrt{35}} & \frac{2}{\sqrt{14}} \end{bmatrix}.$$

With a bit of effort, we can finally confirm that

$$U^H A U = \begin{bmatrix} -1 & 0 \\ 0 & 6 \end{bmatrix},$$

as expected.

5 · Change of Basis
5.1 · The matrix of a linear transformation

Exercise 5.1.1

Solution.   *It's clear that $C_B(\mathbf{0}) = \mathbf{0}$, since the only way to write the zero vector in V in terms of B (or, indeed, any independent set) is to set all the scalars equal to zero.*
    If we have two vectors $\mathbf{v}, \mathbf{w}$ given by

$$\mathbf{v} = a_1\mathbf{e}_1 + \cdots + a_n\mathbf{e}_n$$
$$\mathbf{w} = b_1\mathbf{e}_1 + \cdots + b_n\mathbf{e}_n,$$

then
$$\mathbf{v} + \mathbf{w} = (a_1 + b_1)\mathbf{e}_1 + \cdots + (a_n + b_n)\mathbf{e}_n,$$

so

$$C_B(\mathbf{v} + \mathbf{w}) = \begin{bmatrix} a_1 + b_1 \\ \vdots \\ a_n + b_n \end{bmatrix}$$

$$= \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$= C_B(\mathbf{v}) + C_B(\mathbf{w}).$$

Finally, for any scalar c, we have

$$C_B(c\mathbf{v}) = C_B((ca_1)\mathbf{e}_1 + \cdots + (ca_n)\mathbf{e}_n)$$

$$= \begin{bmatrix} ca_1 \\ \vdots \\ ca_n \end{bmatrix}$$

$$= c \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

$$= cC_B(\mathbf{v}).$$

This shows that C_B is linear. To see that C_B is an isomorphism, we can simply note that C_B takes the basis B to the standard basis of $\mathbb{R}^n$. Alternatively, we can give the inverse: $C_B^{-1} : \mathbb{R}^n \to V$ is given by

$$C_B^{-1} \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = c_1\mathbf{e}_1 + \cdots + c_n\mathbf{e}_n.$$

Exercise 5.1.3

Solution.   *We have*

$$T(1) = (1, 0) = 1(1, 0) + 0(1, -1)$$
$$T(1 - x) = (1, -2) = -1(1, 0) + 2(1, -1)$$
$$T((1 - x)^2) = T(1 - 2 + x^2) = (2, -4) = -2(1, 0) + 4(1, -1).$$

Thus,

$$M_{DB}(T) = \begin{bmatrix} C_D(T(1)) & C_D(T(1 - x)) & C_D(T((1 - x)^2)) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -1 & -2 \\ 0 & 2 & 4 \end{bmatrix}.$$

To confirm, note that

$$M_{DB}(T)C_B(a + bx + cx^2)$$
$$= M_{DB}(T)C_B((a + b + c) - (b + 2c)(1 - x) + c(1 - x)^2)$$
$$= \begin{bmatrix} 1 & -1 & -2 \\ 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} a + b + c \\ -b - 2c \\ c \end{bmatrix}$$

$$\begin{bmatrix} (a + b + c) + (b + 2c) - 2c \\ 0 - 2(b + 2c) + 4c \end{bmatrix} = \begin{bmatrix} a + 2b + c \\ -2b \end{bmatrix},$$

while on the other hand,

$$\begin{aligned} C_D(T(a + bx + cx^2)) &= C_D(a + c, 2b) \\ &= C_D((a + 2b + c)(1, 0) - 2b(1, -1)) \\ &= \begin{bmatrix} a + 2b + c \\ -2b \end{bmatrix}. \end{aligned}$$

Exercise 5.1.7

Solution. *We must first write our general input in terms of the given basis. With respect to the standard basis*

$$B_0 = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right\},$$

we have the matrix $P = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, representing the change from the basis B the basis B_0. The basis D of $P_2(\mathbb{R})$ is already the standard basis, so we need the matrix $M_{DB}(T)P^{-1}$:

```
from sympy import Matrix, init_printing
init_printing()
M = Matrix(3,4,[2,-1,0,3,0,4,-5,1,-1,0,3,-2])
P = Matrix(4,4,[1,0,0,1,0,1,1,0,0,0,1,0,0,1,0,1])
M*P**-1
```

```
\[\bbm 2\amp -2\amp 2\amp 1\\0\amp 3\amp -8\amp 1\\-1\amp
    1\amp 2\amp -1\ebm\]
```

For a matrix $X = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ we find

$$M_{DB}(T)P^{-1}C_{B_0}(X) = \begin{bmatrix} 2 & -2 & 2 & 1 \\ 0 & 3 & -8 & 1 \\ -1 & 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2a - 2b + 2c + d \\ 3b - 8c + d \\ -a + b + 2c - d \end{bmatrix}.$$

But this is equal to $C_D(T(X))$, so

$$\begin{aligned} T\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) &= C_D^{-1} \begin{bmatrix} 2a - 2b + 2c + d \\ 3b - 8c + d \\ -a + b + 2c - d \end{bmatrix} \\ &= (2a - 2b + 2c + d) + (3b - 8c + d)x + (-a + b + 2c - d)x^2. \end{aligned}$$

5.2 · The matrix of a linear operator

Exercise 5.2.4

Solution. *With respect to the standard basis, we have*

$$M_0 = M_{B_0}(T) = \begin{bmatrix} 3 & -2 & 4 \\ 1 & -5 & 0 \\ 0 & 2 & -7 \end{bmatrix},$$

and the matrix P is given by $P = \begin{bmatrix} 1 & 3 & 1 \\ 2 & -1 & 2 \\ 0 & 2 & -5 \end{bmatrix}$. Thus, we find

$$M_B(T) = P^{-1}M_0P = \begin{bmatrix} 9 & 56 & 36 \\ 7 & 15 & 15 \\ -10 & -46 & -33 \end{bmatrix}.$$

5.6 · Jordan Canonical Form

Exercise 5.6.7

Solution.   With respect to the standard basis of $\mathbb{R}^4$, the matrix of T is

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix}.$$

We find (perhaps using the Sage cell provided below, and the code from the example above) that

$$c_T(x) = (x-1)^3(x-2),$$

so T has eigenvalues 1 (of multiplicity 3), and 2 (of multiplicity 1).

  We tackle the repeated eigenvalue first. The reduced row-echelon form of $M - I$ is given by

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

so

$$E_1(M) = span\{\mathbf{x}_1\}, \ \text{where } \mathbf{x}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We now attempt to solve $(M - I)\mathbf{x} = \mathbf{x}_1$. We find

$$\begin{pmatrix} 0 & 1 & 0 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & | & 0 \\ 0 & -1 & 1 & 0 & | & 0 \\ 1 & -1 & 1 & 0 & | & 1 \end{pmatrix} \xrightarrow{RREF} \begin{pmatrix} 1 & 0 & 0 & 0 & | & 1 \\ 0 & 1 & 0 & 0 & | & 0 \\ 0 & 0 & 1 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & | & 0 \end{pmatrix},$$

so $\mathbf{x} = t\mathbf{x}_1 + \mathbf{x}_2$, where $\mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$. We take $\mathbf{x}_2$ as our first generalized eigenvector.

Note that $(M - I)^2\mathbf{x}_2 = (M - I)\mathbf{x}_1 = \mathbf{0}$, so $\mathbf{x}_2 \in null(M - I)^2$, as expected.

  Finally, we look for an element of $null(M - I)^3$ of the form $\mathbf{x}_3$, where $(M - I)\mathbf{x}_3 = \mathbf{x}_2$. We set up and solve the system $(M - I)\mathbf{x} = \mathbf{x}_2$ as follows:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & | & 1 \\ 0 & 0 & 0 & 0 & | & 0 \\ 0 & -1 & 1 & 0 & | & 0 \\ 1 & -1 & 1 & 0 & | & 0 \end{pmatrix} \xrightarrow{RREF} \begin{pmatrix} 1 & 0 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & 0 & | & 1 \\ 0 & 0 & 1 & 0 & | & 1 \\ 0 & 0 & 0 & 0 & | & 0 \end{pmatrix},$$

so $\mathbf{x} = t\mathbf{x}_1 + \mathbf{x}_3$, where $\mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$

Finally, we deal with the eigenvalue 2. *The reduced row-echelon form of* $M - 2I$
is

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

so

$$E_2(M) = span\{\mathbf{y}\}, \text{ where } \mathbf{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Our basis of column vectors is therefore $B = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}\}$. *Note that by design,*

$$M\mathbf{x}_1 = \mathbf{x}_1$$
$$M\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{x}_2$$
$$M\mathbf{x}_3 = \mathbf{x}_2 + \mathbf{x}_3$$
$$M\mathbf{y} = 2\mathbf{y}.$$

The corresponding Jordan basis for $\mathbb{R}^4$ *is*

$$\{(0,0,0,1), (1,0,0,0), (0,1,1,0), (0,0,1,1)\},$$

and with respect to this basis, we have

$$M_B(T) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}.$$