L04VectorEquations

January 31, 2021

```
[1]: # for conversion to PDF use these settings
     # %matplotlib inline
     # qr_setting = 'url'
     # qrviz_setting = 'show'
     #
     # for lecture use notebook
     %matplotlib inline
     qr_setting = None
     #
     %config InlineBackend.figure_format='retina'
     # import libraries
     import numpy as np
     import matplotlib as mp
     import pandas as pd
     import matplotlib.pyplot as plt
     import laUtilities as ut
     import slideUtilities as sl
     import demoUtilities as dm
     import pandas as pd
     from importlib import reload
     from datetime import datetime
     from IPython.display import Image
     from IPython.display import display_html
     from IPython.display import display
     from IPython.display import Math
     from IPython.display import Latex
     from IPython.display import HTML;
```

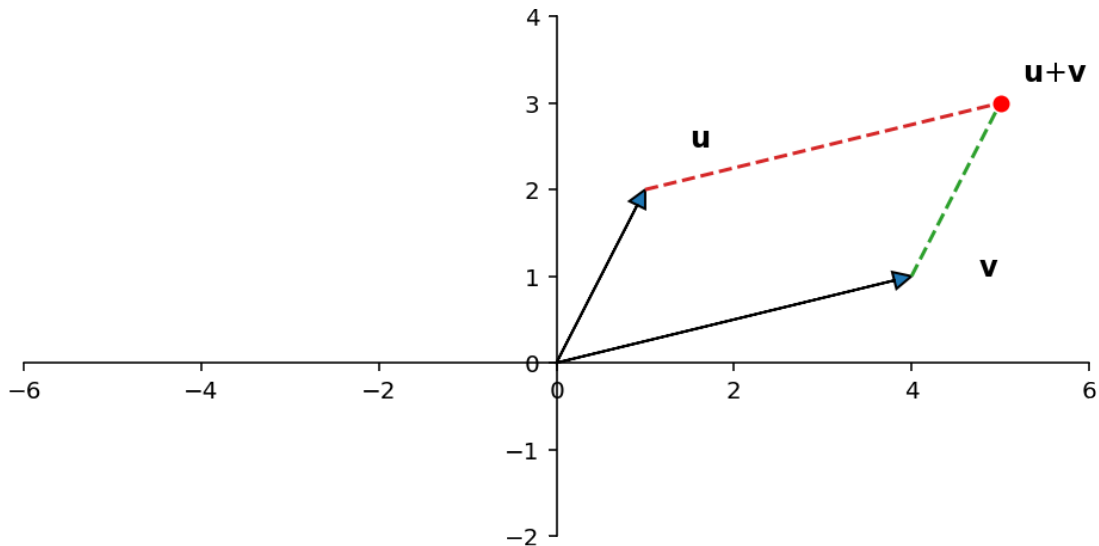## 1 Vector Equations

```
[2]: ax = ut.plotSetup(size=(8,4))
     ut.centerAxes(ax)
     ax.arrow(0, 0, 1, 2, head_width=0.2, head_length=0.2, length_includes_head =␣
      ↪True)
```

```
ax.arrow(0, 0, 4, 1, head_width=0.2, head_length=0.2, length_includes_head =␣
  ↪True)
ax.plot([4,5],[1,3],'--')
ax.plot([1,5],[2,3],'--')
ax.text(5.25,3.25,r'${\bf u}$+${\bf v}$',size=12)
ax.text(1.5,2.5,r'${\bf u}$',size=12)
ax.text(4.75,1,r'${\bf v}$',size=12)
ut.plotPoint(ax,5,3)
ax.plot(0,0,'');
```


1.1 Vectors

We're going to expand our view of what a linear system can represent.

Instead of thinking of it as a collection of equations, we are going to think about it as a **single** equation.

This is a major shift in perspective that will open up an entirely new way of thinking about matrices.

(…and it's going to open the door to computer graphics, machine learning, and statistics … later on!)

To make this fundamental shift we need to introduce the idea of a **vector**. We'll mainly talk about vectors today.

We'll then return to thinking about a linear system – now interpreted as a *vector* equation – in the next lecture.

A matrix with only one column is called a **column vector**, or simply a **vector**.

Here are some examples.

These are vectors in $\mathbb{R}^2$:

$$\mathbf{u} = \begin{bmatrix} 3 \\ -1 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} .2 \\ .3 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

These are vectors in $\mathbb{R}^3$:

$$\mathbf{u} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

1.2 Vectors are Fundamental Objects

We are going to define operations over vectors, so that we can write equations in terms of vectors.

In particular, we now define how to compare vectors, add vectors, and multiply vectors **by a scalar**.

First: Two vectors are **equal** if and only if their corresponding entries are equal.

Thus $\begin{bmatrix} 7 \\ 4 \end{bmatrix}$ and $\begin{bmatrix} 4 \\ 7 \end{bmatrix}$ are **not** equal.

Next: Multiplying a vector by a scalar is accomplished by multiplying each entry by the scalar.

For example:

$$3 \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \end{bmatrix}$$

And finally: the sum of two vectors is the vector whose entries are the corresponding sums.

For example:

$$\begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1+2 \\ -2+5 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

Note that **vectors of different sizes cannot be compared or added.**

For example, if $\mathbf{u} \in \mathbb{R}^2$ and $\mathbf{v} \in \mathbb{R}^3$: * we cannot ask whether $\mathbf{u} = \mathbf{v}$, and * $\mathbf{u} + \mathbf{v}$ is undefined.

So with these three definitions, we have all the tools to write equations using vectors.

For example, if $\mathbf{u} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 2 \\ -5 \end{bmatrix}$ then

$$4\mathbf{u} - 3\mathbf{v} = \begin{bmatrix} -2 \\ 7 \end{bmatrix}$$

1.3 Vectors Correspond to Points

As already noted, an ordered sequence of n numbers can be thought of as a point in $\mathbb{R}^n$. Hence, a vector like $\begin{bmatrix} -2 \\ -1 \end{bmatrix}$ (also denoted $(-2, -1)$) can be thought of as a point on the plane.

```
[3]: ax = ut.plotSetup(size=(6,3))
     ut.centerAxes(ax)
     ut.plotPoint(ax, -2, -1)
     ut.plotPoint(ax, 2, 2)
     ut.plotPoint(ax, 3, -1)
     ax.plot(0, -2, '')
     ax.plot(-4, 0, '')
     ax.text(3.5, -1.1, '$(3,-1)$', size=12)
     ax.text(2.5, 1.9, '$(2,2)$', size=12)
     ax.text(-4.5, -1.1, '$(-2,-1)$', size=12);
```
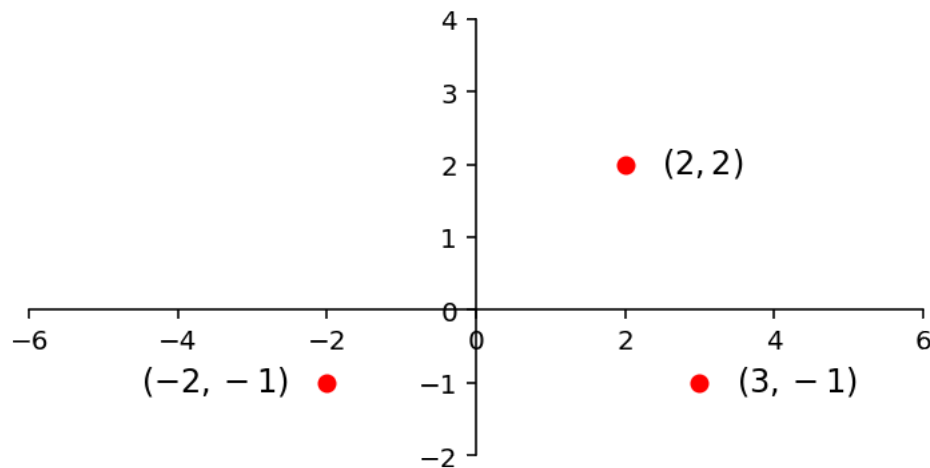

Sometimes we draw an arrow from the origin to the point.

This style comes from physics, but can be a helpful visualization in any case.

```
[4]: ax = ut.plotSetup(size=(6,3))
     ut.centerAxes(ax)
     ax.arrow(0, 0, -2, -1, head_width=0.2, head_length=0.2, length_includes_head =␣
      ↪True)
     ax.arrow(0, 0, 2, 2, head_width=0.2, head_length=0.2, length_includes_head =␣
      ↪True)
     ax.arrow(0, 0, 3, -1, head_width=0.2, head_length=0.2, length_includes_head =␣
      ↪True)
     ax.plot(0, -2, '')
     ax.plot(-4, 0, '')
```

```
ax.plot(0, 2, '')
ax.plot(4 ,0, '')
ax.text(3.5, -1.1, '$(3,-1)$', size=12)
ax.text(2.5, 1.9, '$(2,2)$', size=12)
ax.text(-4.5, -1.1, '$(-2,-1)$', size=12);
```
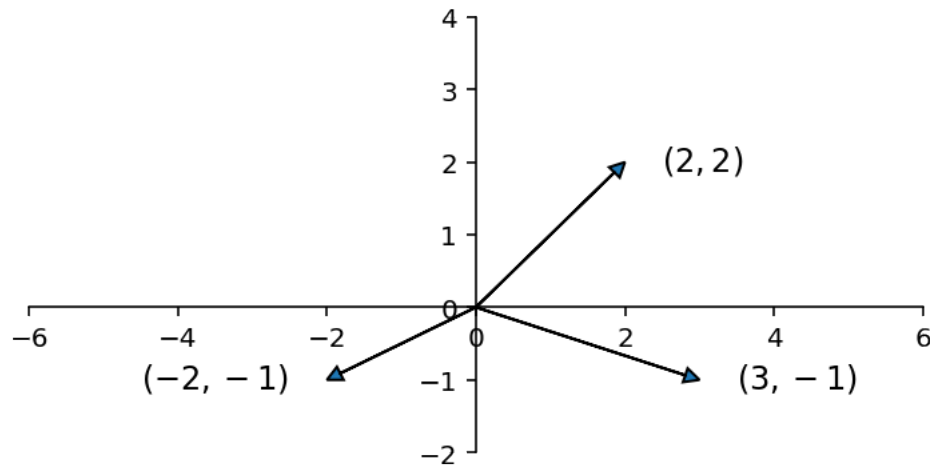

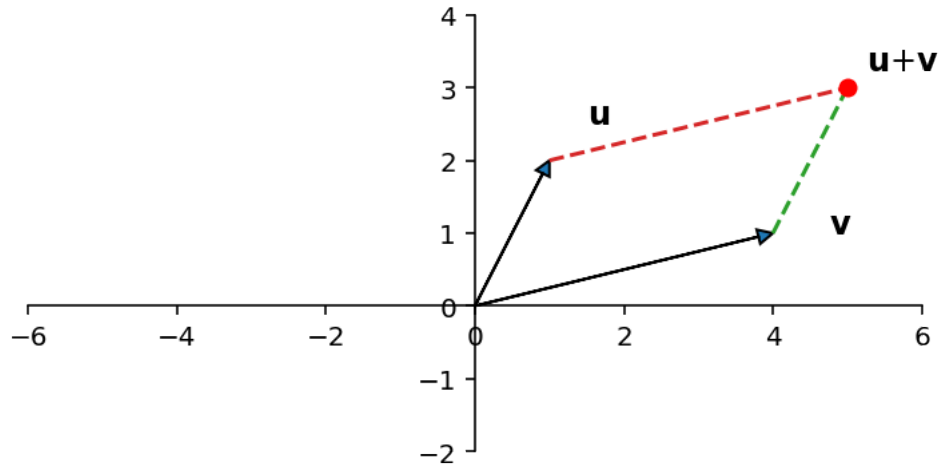## 1.4   Vector Addition, Geometrically

A geometric interpretation of vector sum is as a parallelogram.

If $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^2$ are represented as points in the plane, then $\mathbf{u} + \mathbf{v}$ corresponds to the fourth vertex of the parallelogram whose other vertices are $\mathbf{u}, 0$, and $\mathbf{v}$.

```
[5]: ax = ut.plotSetup(size=(6,3))
     ut.centerAxes(ax)
     ax.arrow(0, 0, 1, 2, head_width=0.2, head_length=0.2, length_includes_head =␣
       ↪True)
     ax.arrow(0, 0, 4, 1, head_width=0.2, head_length=0.2, length_includes_head =␣
       ↪True)
     ax.plot([4,5],[1,3],'--')
     ax.plot([1,5],[2,3],'--')
     ax.text(5.25,3.25,r'${\bf u}$+${\bf v}$',size=12)
     ax.text(1.5,2.5,r'${\bf u}$',size=12)
     ax.text(4.75,1,r'${\bf v}$',size=12)
     ut.plotPoint(ax,5,3)
     ax.plot(0,0,'');
```

5

This should be clear from the definition of vector addition (i.e., addition of corresponding elements).

1.5 Vector Scaling, Geometrically

For a given vector **v** and a scalar a, multiplying a and **v** corresponds to *lengthening* **v** by a *factor* of a.

So 2**v** is twice as long as **v**.

Multiplying by a negative value reverses the "direction" of **v**.

```
[6]: ax = ut.plotSetup(size=(6,3))
     ut.centerAxes(ax)
     factors = [-1.5, 0, 1, 2]
     ftext = ['-3/2', '0', '', '2']
     for f in factors:
         ut.plotPoint(ax, 3.0*f, f)
     ax.arrow(0,0,6,2,head_width=0.2, head_length=0.2, length_includes_head = True)
     ax.arrow(0,0,-4.5,-1.5,head_width=0.2, head_length=0.2,length_includes_head =␣
      ↪True)
     for f in range(len(factors)):
         ax.text(3.0*factors[f]+0.2, factors[f]-0.5, r'{}$\bf v$'.format(ftext[f]),␣
      ↪size=20)
     print('');
```
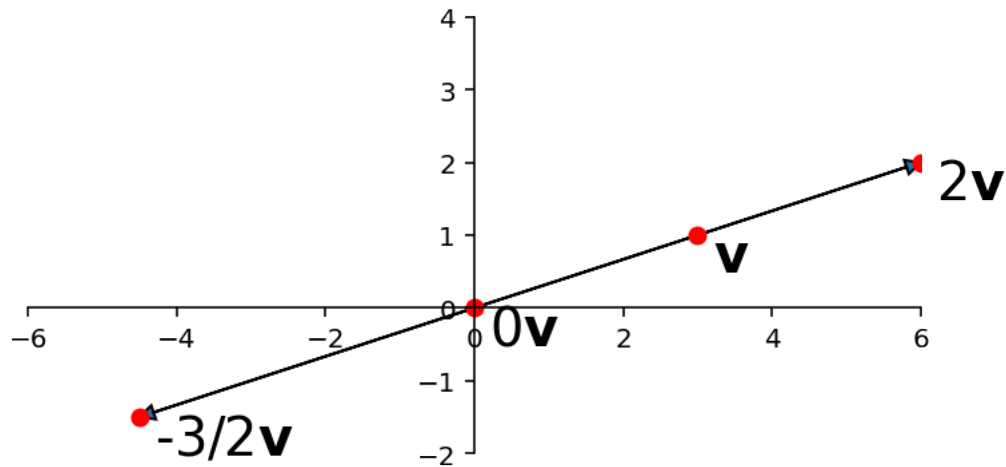
```
[7]: fig = ut.three_d_figure((4, 1), 'Multiplying a vector by a scalar', 0, 4, 0, 6,␣
     ↪0, 8, qr=qr_setting)
     fig.plotCube([2,2,3])
     fig.plotCube([4,4,6])
     fig.text(2, 2, 3, r'$\bf v$', 'v', size=20)
     fig.text(4.1, 4.1, 6.1, r'$2\bf v$', '2v', size=20, color='k')
     fig.plotLine([[0, 0, 0], [4, 4, 6]], 'r', '--')
     fig.plotPoint(4, 4, 6, 'r')
     fig.plotPoint(2, 2, 3, 'r')
     fig.set_title('Multiplying a 3D vector by a scalar')
     fig.save();
```

Figure 4.1
Multiplying a 3D vector by a scalar
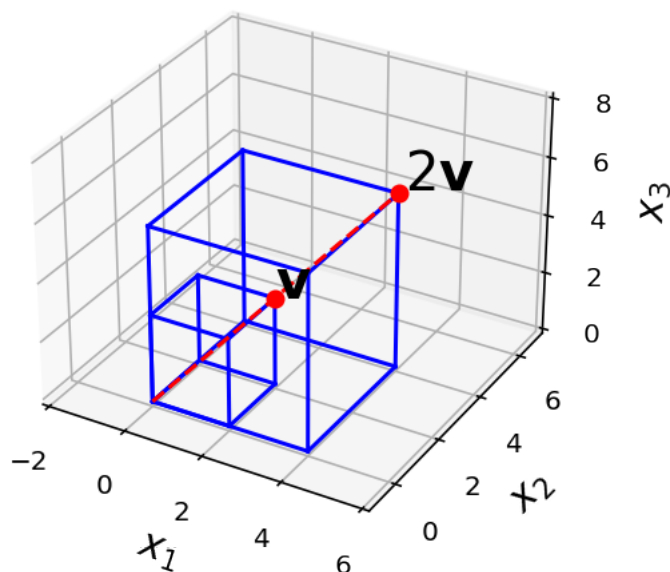
1.6 Question Time! Q4.1

1.7 The Algebra of $\mathbb{R}^n$

We've defined a new object (the vector), and it has certain algebraic properties.

1. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
2. $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
3. $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$
4. $\mathbf{u} + (-\mathbf{u}) = -\mathbf{u} + \mathbf{u} = \mathbf{0}$
5. $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$
6. $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$
7. $c(d\mathbf{u}) = (cd)\mathbf{u}$
8. $1\mathbf{u} = \mathbf{u}$

You can verify each of these by from the definitions of vector addition and scalar-vector multiplication.

1.8 Linear Combinations

A very fundamental thing to do is to construct **linear combinations** of vectors:

$$\mathbf{y} = c_1 \mathbf{v_1} + ... + c_p \mathbf{v_p}$$

8

The c_i values are called **weights**. Weights can be any real number, including zero. So some examples of linear combinations are:

$$\sqrt{3}\mathbf{v_1} + \mathbf{v_2},$$

$$\frac{1}{2}\mathbf{v_1} \;\; (= \frac{1}{2}\mathbf{v_1} + 0\mathbf{v_2})$$

and

$$\mathbf{0} \;\; (= 0\mathbf{v_1} + 0\mathbf{v_2})$$

1.9 Linear Combinations, Geometrically

```
[8]: ax = ut.plotSetup(-6, 6, -4, 4, size=(10,5))
ut.centerAxes(ax)
ax.set_xticks([])
ax.set_yticks([])
factors = [-3, -2, -1, 0, 1, 2, 3]
u = [2, 1]
v = [-1, 1]
for f in factors:
    ut.plotPoint(ax, f*u[0], f*u[1])
    ax.text(f*u[0]+0.25,f*u[1]-0.25,r'{}$\bf u$'.format(f),size=12)
    ut.plotPoint(ax, f*v[0], f*v[1])
    ax.text(f*v[0]+0.25,f*v[1]-0.25,r'{}$\bf v$'.format(f),size=12)
ut.plotPoint(ax,2*u[0]+v[0],2*u[1]+v[1],'g')
ax.text(2*u[0]+v[0]+0.25,2*u[1]+v[1]+0.25,r'$\bf w$',size=12)
ut.plotPoint(ax,-u[0]-v[0],-u[1]-v[1],'b')
ax.text(-u[0]-v[0]+0.25,-u[1]-v[1]+0.25,r'$\bf p$',size=12)
ut.plotPoint(ax,-u[0]+3*v[0],-u[1]+3*v[1],'y')
ax.text(-u[0]+3*v[0]+0.25,-u[1]+3*v[1]+0.25,r'$\bf q$',size=12);
```
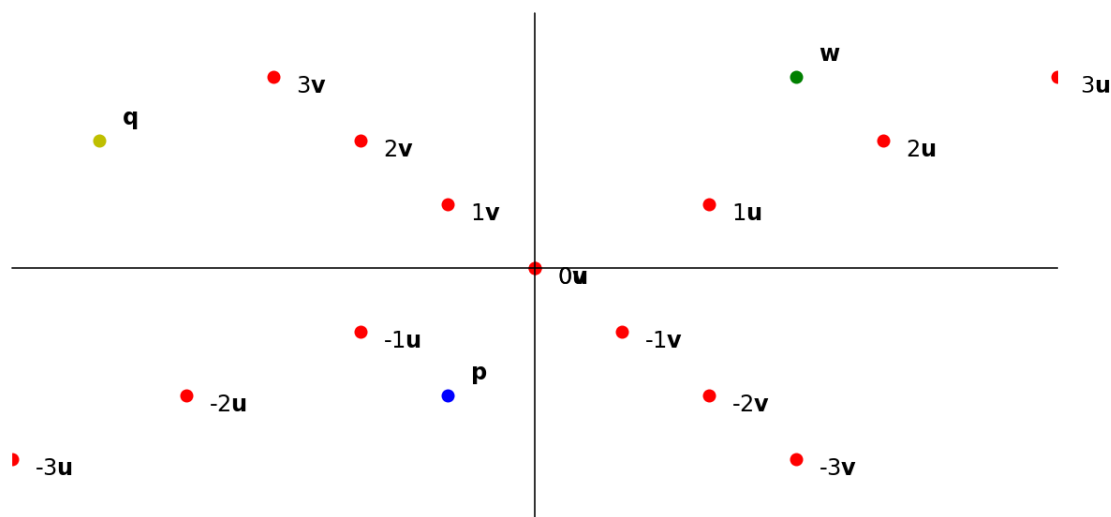
$$\mathbf{w} = 2\mathbf{u} + \mathbf{v}$$

$$\mathbf{p} = -\mathbf{u} - \mathbf{v}$$

$$\mathbf{q} = -\mathbf{u} + 3\mathbf{v}$$

```
[9]: ax = ut.plotSetup(-6, 6, -4, 4, size=(10, 5))
     ut.centerAxes(ax)
     ax.set_xticks([])
     ax.set_yticks([])
     factors = [-3, -2, -1, 0, 1, 2, 3]
     u = [2, 1]
     v = [-1, 1]
     for f in factors:
         ut.plotPoint(ax, f*u[0], f*u[1])
         ax.text(f*u[0]+0.25,f*u[1]-0.25,r'{}$\bf u$'.format(f),size=12)
         ut.plotPoint(ax, f*v[0], f*v[1])
         ax.text(f*v[0]+0.25,f*v[1]-0.25,r'{}$\bf v$'.format(f),size=12)
     ax.
     ↪plot([factors[0]*u[0],factors[-1]*u[0]],[factors[0]*u[1],factors[-1]*u[1]],'b--')
     ax.
     ↪plot([factors[0]*v[0],factors[-1]*v[0]],[factors[0]*v[1],factors[-1]*v[1]],'b--')
     ut.plotPoint(ax,2*u[0]+v[0],2*u[1]+v[1],'g')
     ax.text(2*u[0]+v[0]+0.25,2*u[1]+v[1]+0.25,r'${\bf w} = 2{\bf u} + {\bf␣
     ↪v}$',size=12)
     ut.plotPoint(ax,-u[0]-v[0],-u[1]-v[1],'b')
     ax.text(-u[0]-v[0]-1.2,-u[1]-v[1]-0.5,r'${\bf p} = -{\bf u} -{\bf v}$',size=12)
     ut.plotPoint(ax,-u[0]+3*v[0],-u[1]+3*v[1],'y')
     ax.text(-u[0]+3*v[0]-0.25,-u[1]+3*v[1]+0.35,r'${\bf q} = -{\bf u} + 3{\bf␣
     ↪v}$',size=12);
```
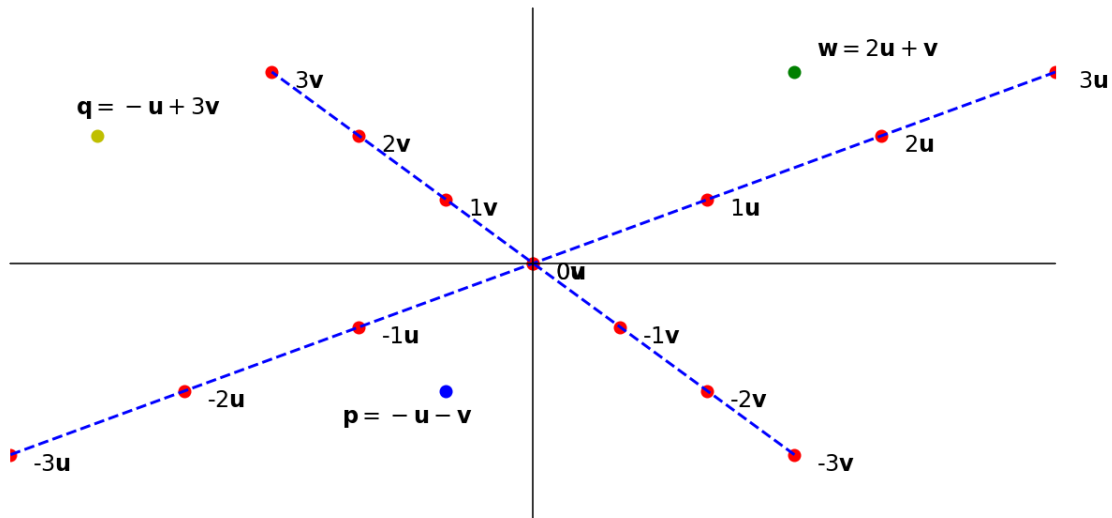
1.10 Question Time! Q4.2

1.11 A Fundamental Question

We are now going to take up a very basic question that will lead us to a deeper understanding of linear systems.

> Given some set of vectors $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_k}$, can a given vector $\mathbf{b}$ be written as a linear combination of $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_k}$?

Let's take a specific example.

Let $\mathbf{a_1} = \begin{bmatrix} 1 \\ -2 \\ -5 \end{bmatrix}$, $\mathbf{a_2} = \begin{bmatrix} 2 \\ 5 \\ 6 \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} 7 \\ 4 \\ -3 \end{bmatrix}$.

We need to determine whether $\mathbf{b}$ can be generated as a linear combination of $\mathbf{a_1}$ and $\mathbf{a_2}$. That is, we seek to find whether weights x_1 and x_2 exist such that

$$x_1 \mathbf{a_1} + x_2 \mathbf{a_2} = \mathbf{b}.$$

Solution. We are going to convert from a **single** vector equation to a **set** of linear equations, that is, a **linear system.**

We start by writing the form of the solution, if it exists:

$$x_1 \mathbf{a_1} + x_2 \mathbf{a_2} = \mathbf{b}.$$

Written out, this is:

11

$$x_1 \begin{bmatrix} 1 \\ -2 \\ -5 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ -3 \end{bmatrix}.$$

By the definition of scalar-vector multiplication, this is:

$$\begin{bmatrix} x_1 \\ -2x_1 \\ -5x_1 \end{bmatrix} + \begin{bmatrix} 2x_2 \\ 5x_2 \\ 6x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ -3 \end{bmatrix}.$$

By the definition of vector addition, this is:

$$\begin{bmatrix} x_1 + 2x_2 \\ -2x_1 + 5x_2 \\ -5x_1 + 6x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ -3 \end{bmatrix}.$$

By the definition of vector equality, this is:

$$\begin{aligned} x_1 + 2x_2 &= 7 \\ -2x_1 + 5x_2 &= 4 \\ -5x_1 + 6x_2 &= -3 \end{aligned}.$$

We know how to solve this! Firing up Gaussian Elimination, we first construct the augmented matrix of this system, and then find its reduced row echelon form:

$$\begin{bmatrix} 1 & 2 & 7 \\ -2 & 5 & 4 \\ -5 & 6 & -3 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 7 \\ 0 & 9 & 18 \\ 0 & 16 & 32 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 7 \\ 0 & 1 & 2 \\ 0 & 16 & 32 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

Voila!

Now, reading off the answer, we have $x_1 = 3$, $x_2 = 2$. So we have found the solution to our original problem:

$$3 \begin{bmatrix} 1 \\ -2 \\ -5 \end{bmatrix} + 2 \begin{bmatrix} 2 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ -3 \end{bmatrix}.$$

In other words, we have found that
$$3\mathbf{a_1} + 2\mathbf{a_2} = \mathbf{b}.$$

1.12 A Vector Equation is a Linear System!

And vice versa!

Let's state this formally. First, of all, recalling that vectors are columns, we can write the augmented matrix for the linear system in a very simple way.

For the vector equation

$$x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b},$$

the corresponding linear system has augmented matrix:

$$[\mathbf{a_1} \ \mathbf{a_2} \ \mathbf{b}].$$

Then we can make the following statement:

A vector equation

$$x_1\mathbf{a_1} + x_2\mathbf{a_2} + ... + x_n\mathbf{a_n} = \mathbf{b}$$

has the same solution set as the linear system whose augmented matrix is

$$[\mathbf{a_1} \ \mathbf{a_2} \ ... \ \mathbf{a_n} \ \mathbf{b}].$$

This is a powerful concept; we have related

- a single equation involving *columns*
- to a set of equations corresponding to *rows*.

1.13 Question Time! Q4.3

1.14 Span

If a vector equation is equivalent to a linear system, then it must be possible for a vector equation to be inconsistent as well.

How can we understand what it means – in terms of *vectors* – for a vector equation to be inconsistent?

The answer involves a new concept: the **span** of a set of vectors.

Let's say we are given vectors $\mathbf{a_1}, \mathbf{a_2}$, and $\mathbf{b}$.

And, say we know that it is possible to express $\mathbf{b}$ as a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$.

That is, there are some x_1, x_2 such that $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$.

Then we say that $\mathbf{b}$ is in the **Span** of the set of vectors $\{\mathbf{a_1}, \mathbf{a_2}\}$.

More generally, let's say we are given a set of vectors $\mathbf{v_1}, ..., \mathbf{v_p}$ where each $\mathbf{v_i} \in \mathbb{R}^n$.

Then the set of all linear combinations of $\mathbf{v_1}, ..., \mathbf{v_p}$ is denoted by

$$\text{Span}\{\mathbf{v_1}, ..., \mathbf{v_p}\}$$

and is called the **subset of $\mathbb{R}^n$ spanned by $\mathbf{v_1}, ..., \mathbf{v_p}$**.

Span of a single vector in $\mathbb{R}^3$

```
[10]: fig = ut.three_d_figure((5, 5), 'Span of a single vector', -10, 10, -10, 10,␣
      ↪-10, 10, qr = qr_setting)
      v = [4,4,2]
      fig.text(v[0], v[1], v[2], r'$\bf v$', 'v', size=20)
      fig.text(-9, -7, -9, r'Span{$\bf v$}', 'Span{v}', size=16)
      fig.text(0.2, 0.2, -4, r'$\bf 0$', '0', size=20)
      # plotting the span of v
      # this is based on the reduced echelon matrix that expresses the system whose␣
      ↪solution is v
      fig.plotIntersection([1, 0, -v[0]/v[2], 0], [0, 1, -v[1]/v[2], 0], '--', 'Red')
      fig.plotPoint(v[0], v[1], v[2], 'r')
      fig.plotPoint(0, 0, 0, 'b')
      # plotting the axes
      fig.plotIntersection([0,0,1,0], [0,1,0,0])
      fig.plotIntersection([0,0,1,0], [1,0,0,0])
      fig.plotIntersection([0,1,0,0], [1,0,0,0])
      fig.set_title('Span of a Single Vector')
      fig.save()
```

Figure 5.5
Span of a Single Vector

14
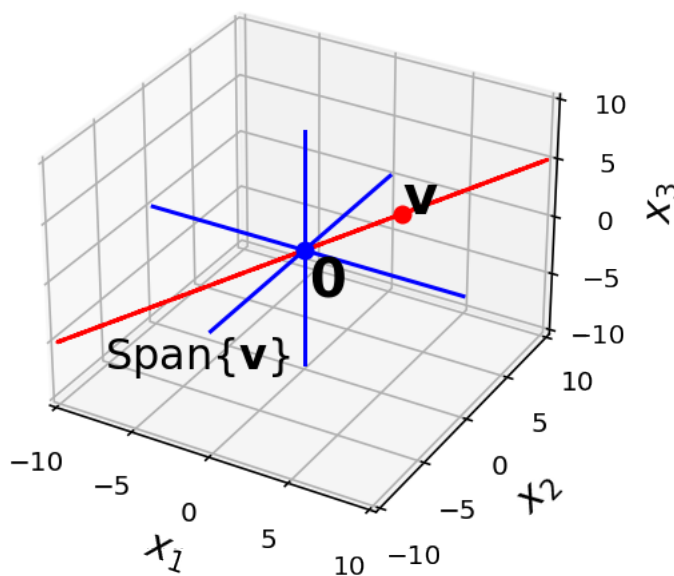
Span of two vectors in $\mathbb{R}^3$

```
[11]: fig = ut.three_d_figure((5, 5), 'Span of two vectors', -10, 10, -10, 10, -10,
      ↪10, qr = qr_setting)
      v = [ 4.0, 4.0, 2.0]
      u = [-4.0, 3.0, 1.0]
      fig.text(v[0], v[1], v[2], r'$\bf v$', 'v', size=20)
      fig.text(u[0], u[1], u[2], r'$\bf u$', 'u', size=20)
      fig.text(1, -4, -10, r'Span{$\bf u,v$}', 'Span{u, v}', size=16)
      fig.text(0.2, 0.2, -4, r'$\bf 0$', '0', size=20)
      # plotting the span of v
      fig.plotSpan(u, v, 'Green')
      fig.plotPoint(u[0], u[1], u[2], 'b')
      fig.plotPoint(v[0], v[1], v[2], 'r')
      fig.plotPoint(0, 0, 0, 'b')
      # plotting the axes
      fig.plotIntersection([0,0,1,0], [0,1,0,0])
      fig.plotIntersection([0,0,1,0], [1,0,0,0])
      fig.plotIntersection([0,1,0,0], [1,0,0,0])
      fig.set_title('Span of two vectors')
      fig.save()
```

Figure 5.5
Span of two vectors
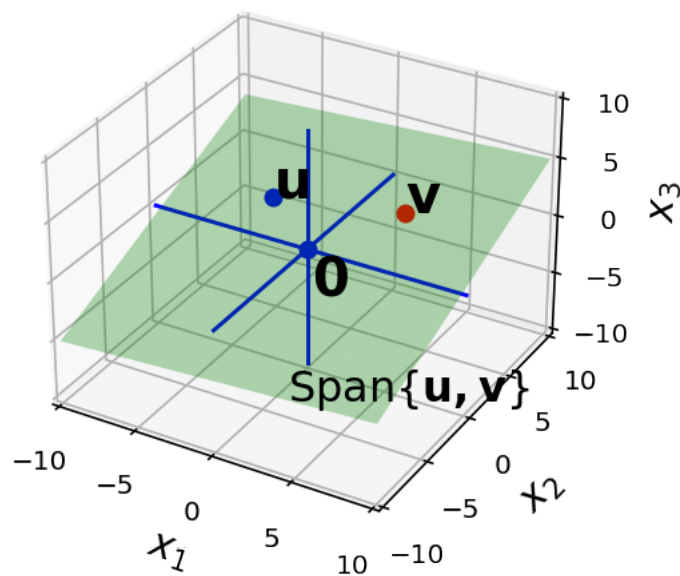
1.15 Asking Whether a Vector Lies Within a Span

Asking whether a vector $\mathbf{b}$ is in Span$\{\mathbf{v_1}, ..., \mathbf{v_p}\}$ is the same as asking whether the vector equation

$$x_1\mathbf{v_1} + x_2\mathbf{v_2} + \cdots + x_p\mathbf{v_p} = \mathbf{b}$$

has a solution.

... which we now know is the same as asking whether the linear system with augmented matrix

$$\begin{bmatrix} \mathbf{v_1} & \mathbf{v_2} & ... & \mathbf{v_p} & \mathbf{b} \end{bmatrix}$$

has a solution.

Let $\mathbf{a_1} = \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}$, $\mathbf{a_2} = \begin{bmatrix} 5 \\ -13 \\ -3 \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} 6 \\ 8 \\ -5 \end{bmatrix}$.

Then Span$\{\mathbf{a_1}, \mathbf{a_2}\}$ is a plane through the origin in $\mathbb{R}^3$. Is $\mathbf{b}$ in that plane?

Solution: Does the equation $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$ have a solution?

To answer this, consider the equivalent linear system. Solve the system by row reducing the augmented matrix

$$\begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \mathbf{b} \end{bmatrix} :$$

$$\begin{bmatrix} 1 & 5 & 6 \\ -2 & -13 & 8 \\ 3 & -3 & -5 \end{bmatrix} \sim \begin{bmatrix} 1 & 5 & 6 \\ 0 & -3 & 20 \\ 0 & -18 & -23 \end{bmatrix} \sim \begin{bmatrix} 1 & 5 & 6 \\ 0 & -3 & 20 \\ 0 & 0 & -143 \end{bmatrix}$$

The third row shows that the system has no solution.

This means that the vector equation $x_1\mathbf{a_1} + x_2\mathbf{a_2} = \mathbf{b}$ has no solution.

So $\mathbf{b}$ is *not* in Span$\{\mathbf{a_1}, \mathbf{a_2}\}$.

What does this situation look like geometrically?

```
[12]: fig = ut.three_d_figure((5, 5), 'Inconsistent System', -15, 15, -15, 15, -15,␣
      ↪15, qr = qr_setting)
      v = [1.0,  -2.0,  3.0]
      u = [5.0, -13.0, -3.0]
      b = [6.0,   8.0, -5.0]
      fig.text(v[0], v[1], v[2], r'$\bf a_1$', 'a_1', size=20)
      fig.text(u[0], u[1], u[2], r'$\bf a_2$', 'a_2', size=20)
      fig.text(b[0], b[1], b[2], r'$\bf b$', 'b', size=20)
      fig.text(0.2, 0.2, -4, r'$\bf 0$', '0', size=20)
      # plotting the span of v
      fig.plotSpan(u, v, 'Green')
      fig.plotPoint(u[0], u[1], u[2], 'r')
```
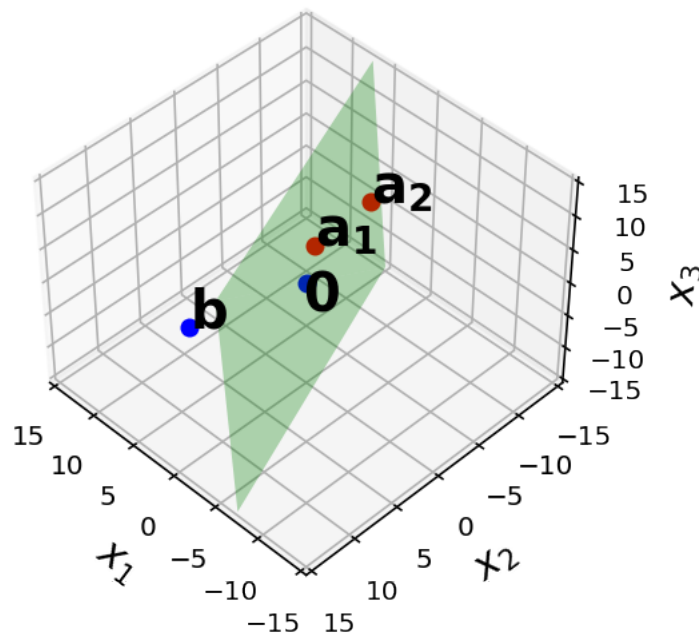
```
fig.plotPoint(v[0], v[1], v[2], 'r')
fig.plotPoint(b[0], b[1], b[2], 'b')
fig.plotPoint(0, 0, 0, 'b')
fig.ax.view_init(azim=135.0,elev=43.0)
fig.set_title('An Inconsistent System')
fig.save()
```

Figure 5.5
An Inconsistent System

1.16 Two Distinct Vector Spaces

Be sure to keep clear in your mind that we have been working with two different vector spaces.

One vector space is for visualizing equations that correspond to rows.

The other vector space is for visualizing vector equations (involving columns).

These are two different ways of visualizing the same linear system.

Let's look at an inconsistent system both ways:

Here are the views, first as a vector equation, and then as a system of equations.

```
[13]: fig = ut.three_d_figure((5, 5), 'Inconsistent Vector Equation',
                               -15, 15, -15, 15, -15, 15, qr = qr_setting)
      v = [1.0,  -2.0,  3.0]
      u = [5.0, -13.0, -3.0]
```

```
b = [6.0,    8.0, -5.0]
fig.text(v[0], v[1], v[2], r'$\bf a_1$', 'a_1', size=20)
fig.text(u[0], u[1], u[2], r'$\bf a_2$', 'a_2', size=20)
fig.text(b[0], b[1], b[2], r'$\bf b$', 'b', size=20)
#ax.text(1,-4,-10,r'Span{$\bf a,b$}',size=16)
fig.text(0.2, 0.2, -4, r'$\bf 0$', '0', size=20)
# plotting the span of v
fig.plotSpan(u, v, 'Green')
fig.plotPoint(u[0], u[1], u[2], 'r')
fig.plotPoint(v[0], v[1], v[2], 'r')
fig.plotPoint(b[0], b[1], b[2], 'b')
fig.plotPoint(0, 0, 0, 'b')
fig.ax.view_init(azim=135.0,elev=43.0)
# plt.suptitle('$x_1{\bf a_1} + x_2{\bf a_2} = {\bf b}$',size=20)
fig.set_title('Inconsistent Vector Equation')
fig.save()
```


Figure 5.5
Inconsistent Vector Equation
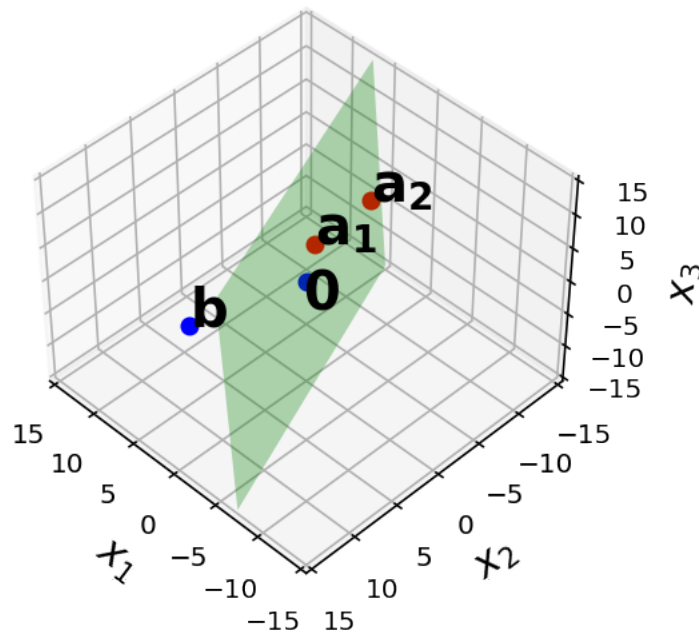
```
[14]: A = np.array([v, u, [0,0,0], b]).T
      eq1 = A[0]
      eq2 = A[1]
      eq3 = A[2]
      fig = ut.three_d_figure((5, 5), 'Inconsistent System of Equations',
                              -15, 15, -15, 15, -15, 15, qr = qr_setting)
```
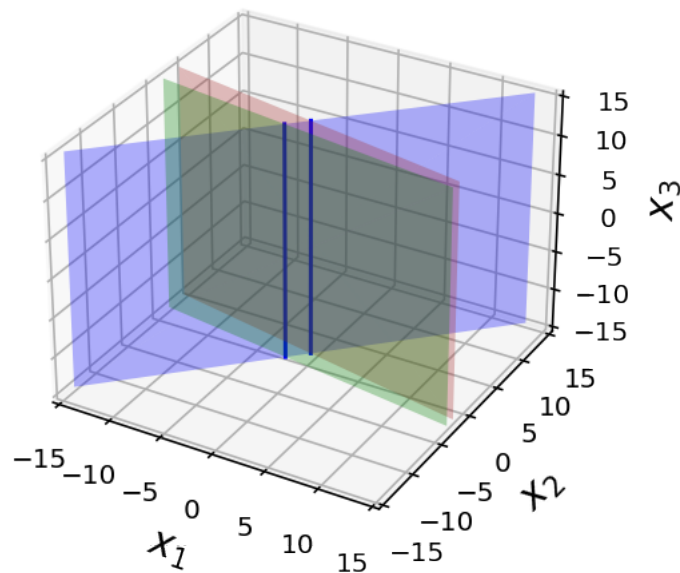
18

```
fig.plotLinEqn(eq1, 'Brown')
fig.plotLinEqn(eq2, 'Green')
fig.plotLinEqn(eq3, 'Blue')
fig.plotIntersection(eq1, eq3, 'Brown')
fig.plotIntersection(eq2, eq3, 'Green')
fig.set_title('Inconsistent System of Equations')
fig.save()
```


Figure 5.5
Inconsistent System of Equations

The second vector space (and visualization) is more familiar to us right now.

However, as the course goes on we will use the first vector space – the one in which columns are vectors – much more often. Make sure you understand the figure!