# AN INTRODUCTION TO LINEAR ALGEBRA USING PYTHON

**Summer 2021**
**Zoom Lecture: Tu: 2:00-4:00 p.m.**
**National Science Foundation (NSF) Center for Integrated Quantum Materials (CIQM), DMR -1231319**
**Dr. Steven L. Richardson (srichards22@comcast.net)**
**Professor Emeritus of Electrical Engineering, Department of Electrical and Computer Engineering, Howard University, Washington, DC**
**and**
**Faculty Associate in Applied Physics, John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA**

## PROBLEM SET VI
**(due Tuesday, June 29, 2021)**

### Problem 1

Please look at the absolutely beautiful animated lecture by Grant Sanderson "Essence of Linear Algebra: Preview (5:04 minutes)." His geometric discussion of the basic concepts in linear algebra is so spectacular and I could never accomplish what he does in a lecture. Learning in the 21$^{th}$ century is accomplished through all sorts of mechanisms (e.g. lectures, problem sets, reading, recitation sections, videos, etc.) so we should take advantage of all of these approaches!

### Problem 2

Please look at the absolutely beautiful animated lecture by Grant Sanderson "Vectors, Essence of Linear Algebra: Chapter 1 (9:51 minutes)." His geometric discussion of the basic concepts in linear algebra is so spectacular and I could never accomplish what he does in a lecture. Learning in the 21$^{th}$ century is accomplished through all sorts of mechanisms (e.g. lectures, problem sets, reading, recitation sections, videos, etc.) so we should take advantage of all of these approaches!

### Problem 3

Please look at the absolutely beautiful animated lecture by Grant Sanderson "Linear Combinations, Span, and Basis Vectors, Essence of Linear Algebra: Chapter 2 (9:58 minutes)." His geometric discussion of the basic concepts in linear algebra is so spectacular and I could never accomplish what he does in a lecture. Learning in the 21$^{th}$ century is accomplished through all sorts of mechanisms (e.g. lectures, problem sets, reading, recitation sections, videos, etc.) so we should take advantage of all of these approaches!

**Problem 4**

The set of all vectors in $\mathbf{R}^3$ that are orthogonal to a nonzero vector is what kind of geometric object?

**Problem 5**

Is the following statement true or false: If $\vec{u}$ is orthogonal to $\vec{v} + \vec{w}$, then $\vec{u}$ is orthogonal to $\vec{v}$ and $\vec{w}$. Please justify your answer.

**Problem 6**

Is the following statement true or false: If $\vec{u}$ is orthogonal to $\vec{v}$ and $\vec{w}$, then $\vec{u}$ is orthogonal to $\vec{v} + \vec{w}$. Please justify your answer.

**Problem 7**

Indicate whether the following statement is true or false and justify your answer: If two vectors $\vec{u}$ and $\vec{v}$ in $\mathbf{R}^2$ are orthogonal to a nonzero vector $\vec{w}$ in $\mathbf{R}^2$, then $\vec{u}$ and $\vec{v}$ are scalar multiples of each other.

**Problem 8**

Draw an arbitrary triangle (not a right triangle). Let the vector $\vec{a}$ describe one side of the triangle whose tail starts at some point on your paper and whose head goes to one of the three vertices of the triangle. Now let the vector $\vec{b}$ describe another side of the triangle whose tail also starts at your chosen point on the paper and whose head goes to the other vertex of the triangle. Finally, let the vector $\vec{c}$ start at the head of $\vec{a}$ and go to the head of $\vec{b}$.

If you define

$$\vec{c} = \vec{b} - \vec{a}$$

prove the Law of Cosines

$$c^2 = a^2 + b^2 - 2ab\cos\theta$$

where $\theta$ is the angle between the vectors $\vec{a}$ and $\vec{b}$. (Hint: Think about taking the dot product of a suitable vector.)

**Problem 9**

Let us take the Law of Cosines from Problem 8 and realize that all of the lengths in this equation can be expressed as the norms of the vectors $\vec{a}$, $\vec{b}$, and $\vec{c}$

$$||\vec{c}||^2 = ||\vec{a}||^2 + ||\vec{b}||^2 - 2||\vec{a}||||\vec{b}|| \cos\theta$$

and

$$||\vec{c}|| = ||\vec{b} - \vec{a}||$$

or

$$||\vec{b} - \vec{a}||^2 = ||\vec{a}||^2 + ||\vec{b}||^2 - 2||\vec{a}||||\vec{b}|| \cos\theta$$

Show that

$$||\vec{b} - \vec{a}||^2 = (\vec{b} - \vec{a})^T(\vec{b} - \vec{a})$$

and use the distributive property of vector multiplication to show

$$||\vec{b} - \vec{a}||^2 = ||\vec{a}||^2 + ||\vec{b}||^2 - 2\vec{a}^T\vec{b}$$

Since we now have two equivalent ways of expressing

$$||\vec{b} - \vec{a}||^2$$

or

$$||\vec{b} - \vec{a}||^2 = ||\vec{a}||^2 + ||\vec{b}||^2 - 2\vec{a}^T\vec{b} = ||\vec{a}||^2 + ||\vec{b}||^2 - 2||\vec{a}||||\vec{b}|| \cos\theta$$

we have discovered both an algebraic and a geometric way of expressing the dot product of two vectors

$$\vec{a}^T\vec{b} = \vec{a} \cdot \vec{b} = ||\vec{a}||||\vec{b}|| \cos\theta.$$

**Problem 10**

Let us take the results of Problem 9 to prove the Cauchy-Schwarz inequality that the magnitude of the dot product of two vectors is less than or equal to the product of the norms of these two vectors

$$|\vec{a} \cdot \vec{b}| = |\vec{a}^T \vec{b}| = ||\vec{a}||||\vec{b}||| \cos\theta|$$

and

$$|\vec{a} \cdot \vec{b}| = |\vec{a}^T \vec{b}| \leq ||\vec{a}||||\vec{b}||$$

**Problem 11**

Let us build the following triangle (not necessarily a right triangle) by starting at some point on the paper and defining a vector $\vec{u}$ that starts at that point with its head ending somewhere in space. Next write down a vector $\vec{v}$ that starts at the head of $\vec{u}$ and goes to some point in space other than your original point. Finally define the vector $\vec{u} + \vec{v}$ using the parallelogram law of addition for vectors. Use the Cauchy-Schwarz inequality to prove the triangle inequality

$$||\vec{u} + \vec{v}|| \leq ||\vec{u}|| + ||\vec{v}||$$

**Problem 12**

Determine if the vectors

$$\vec{\mathbf{v}}_1 = \begin{pmatrix} 1 \\ 1 \\ -i \end{pmatrix}$$

$$\vec{\mathbf{v}}_2 = \begin{pmatrix} 0 \\ i \\ i \end{pmatrix}$$

$$\vec{\mathbf{v}}_3 = \begin{pmatrix} 0 \\ 1 \\ -i \end{pmatrix}$$

are linearly independent.

**Problem 13**

Determine if the vectors

$$\vec{\mathbf{v}}_{\mathbf{1}} = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix}$$

$$\vec{\mathbf{v}}_{\mathbf{2}} = \begin{pmatrix} \mathbf{1} \\ -\mathbf{1} \\ \mathbf{1} \\ -\mathbf{1} \end{pmatrix}$$

$$\vec{\mathbf{v}}_{\mathbf{3}} = \begin{pmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \end{pmatrix}$$

$$\vec{\mathbf{v}}_{\mathbf{4}} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{2} \\ \mathbf{0} \end{pmatrix}$$

are linearly independent.

**Problem 14**

Determine if the vectors

$$\vec{\mathbf{v}}_{\mathbf{1}} = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix}$$

$$\vec{\mathbf{v}}_{\mathbf{2}} = \begin{pmatrix} \mathbf{1} \\ -\mathbf{1} \\ \mathbf{1} \end{pmatrix}$$

$$\vec{\mathbf{v}}_{\mathbf{3}} = \begin{pmatrix} -\mathbf{1} \\ \mathbf{1} \\ -\mathbf{1} \end{pmatrix}$$

are linearly independent.

**Problem 15**

Refer back to Problem 4 of Problem Set V. Consider the coefficient matrix **A** for this problem and see if the column vectors of this matrix are linearly independent. How does your answer relate to the answer of Problem 4 of Problem Set V?

**Problem 16**

Refer back to Problem 5 of Problem Set V. Consider the coefficient matrix **A** for this problem and see if the column vectors of this matrix are linearly independent. How does your answer relate to the answer of Problem 5 of Problem Set V?

**Problem 17**

Refer back to Problem 6 of Problem Set V. Consider the coefficient matrix **A** for this problem and see if the column vectors of this matrix are linearly independent. How does your answer relate to the answer of Problem 6 of Problem Set V?

**Python Exercise 6**

Let us explore some basic ideas of vector analysis using NumPy.

```
import numpy as np
x = np.array([[1],[2],[3]])#Here our vector is expressed in column representation which is the usual convention
x = np.array([[1,2,3]])#Here our vector is expressed in row representation. You see here that it is easier to enter vectors in row representation in Python than in column representation. We will still, however, stick with the convention of using vectors in column representation when doing linear algebra throughout our course and in general.
print(x)
print(2.2*x)# Scalar multiplication of the vector x by 2.2. Note we previously used upper case letters for matrices and we now use lower case letters for vectors. This is just our convention and NumPy does not make this distinction!
print(x + 2)# Addition of a scalar number to a vector
```

```
import numpy as np
x = np.array([-1,2,2])
y = np.array([1,0,-3])
print(np.inner(x,y))#calculating the scalar product or dot product or inner product of two vectors
```

or alternatively

```
import numpy as np
x = np.array([-1,2,2])
y = np.array([1,0,-3])
x@y #another way to find the inner product of two vectors
```

```
import numpy as np
a = np.array([1,2])
b = np.array([3,4])
alpha = -0.5#definition of the constant alpha
beta = 1.5#definition of the constant beta
c = alpha*a + beta*b#forming a linear combination of two vectors using scalar-vector multiplication and vector addition
print(c)
```

We now discuss how to use the wonderful package **matplotlib** in Python to plot useful things in linear algebra.

import numpy as np

from matplotlib import pyplot as plt # The pyplot command is the most important function in the matplotlib library and it is used to plot two-dimensional data. The pyplot command is imported from matplotlib and abbreviated by plt for convenience.

F = [20.1, 20.8, 21.9, 22.5, 22.7, 22.3, 21.8, 21.2, 20.9, 20.1] # Here we define an arbitrary set of ten real numbers known as a **list** in Python. A list in Python has no intrinsic orientation in Python as it is neither a row nor column vector.

print (F)

import matplotlib.pyplot as plt

plt.plot(F) # Note that the counting of integers along the x-axis begins with 0 and not 1 in Python. We will return to this point later! Here we have for the first time used the Python command **plt.plot** to plot something!

Here we plot a function using **matplotlib**:

import numpy as np

import matplotlib.pyplot as plt

y = 2 * x + 5 #this is the function we are plotting

plt.title("Matplotlib demo") #plot labels

plt.xlabel("x axis caption") #plot labels

plt.ylabel("y axis caption") #plot labels

plt.plot(x,y) #Plot our function. Note that if we only wanted to show points and no line we would use the command plt.plot(x,y,"ob")

plt.show() # show our output from the above calculation

Note the values of x along the x-axis. Look up the **plt.plot** in more detail using Google to see what is going on and how you can change these numbers!

Now let us plot a vector using **matplot.lib**:

```
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
# Vector origin location
x = [0]
y = [0]
# Directional vectors
u = [2]
v = [1]
# Creating plot
plt.quiver(x,y,u,v, color=”b”, units=”xy”, scale=1)
plt.title(”Single Vector”)
# x-lim and y-lim
plt.xlim(-2, 5)
plt.ylim(-2, 2.5)
# Show plot with gird
plt.grid()
plt.show()
```

Now let us discover something very interesting in Python

v = np.array ([2, -1]) # Let us create the following vector which is in column representation

v[0]# Let us print out its first coordinate

Note v[0] is the first component of the vector $\boldsymbol{\vec{v}}$. You may ask why is not the first component of $\boldsymbol{\vec{v}}$ simply v[1] ?

Here is the deal. Computer scientists code the entries of a vector differently than scientists and engineers. **They start with the index 0 and not 1.** Now you understand what we were talking about regarding **plt.plot(F)** on Page 8! This extends to matrices also as what we call $\boldsymbol{a_{11}}$ as the top left-hand component of any matrix is what computer scientists call $\boldsymbol{a_{00}}$ where the row index starts at 0 and the column index starts at 0! Now you understand what our mystery was back in Problem Set III when we were visualizing matrices!

Are we going to go back and change our notation in this course? No! What saves us is that this is an internal convention used in Python and it affects none of the information or results we have discussed so far in this course. One does, however, have to be careful in paying attention to this convention especially when writing programs in Python. We are not going to have this problem at least in this introductory course, but you should be aware of this convention in computer science.

1. Use Python to create **five** random vectors in two dimensions and plot them one at a time. Print out the two components of each vector in Python. Also take the dot product of each vector with itself and each of the other four vectors.