vectors_lab

April 30, 2018

1 Vectors Lab

In this notebook you will learn how to graph two dimensional (2D) vectors and certain vector computations.

Specifically: 1. Plotting a 2D vector 2. Multiplying a 2D vector by a scalar and plotting the results 3. Adding two 2D vectors together and plotting the results

For this lab, we will be using the python package NumPy for creating vectors and computing vector operations. For the graphing aspects of the lab, we will be using python package Matplotlib.

1.1 Plotting a Vector in 2D

For this part of the lab, we will plot the vector $\vec{v}$ defined below.

$$\vec{v} = \begin{bmatrix} 1\\ 1 \end{bmatrix}$$

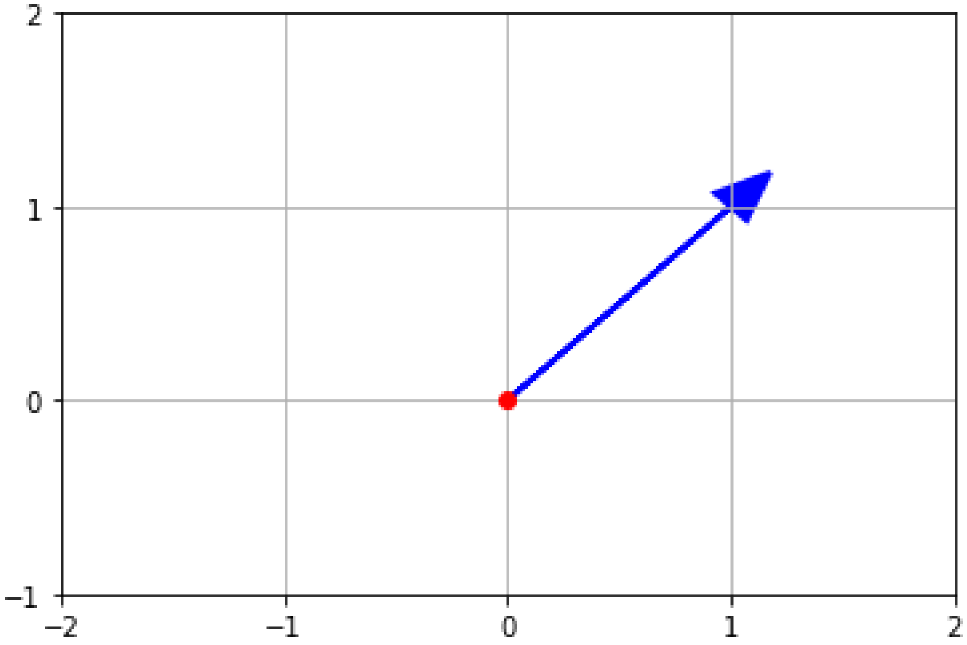
Below is an outline that describes what is included in the Python code to plot vector $\vec{v}$. 1. Make both NumPy and Matlibplot python packages available using the *import* method

2. Define vector $\vec{v}$

3. Plot vector $\vec{v}$ using Matlibplot
1. Create a variable ***ax*** to reference the axes of the plot 2. Plot the origin as a red dot at point 0,0 using ***ax*** and *plot* method 3. Plot vector $\vec{v}$ as a blue arrow with origin at 0,0 using ***ax*** and *arrow* method 4. Format x-axis 1. Set limits using *xlim* method 2. Set major tick marks using ***ax*** and *set_xticks* method 5. Format y-axis 1. Set limits using *ylim* method 2. Set major tick marks using ***ax*** and *set_yticks* method
6. Create the gridlines using *grid* method
7. Display the plot using *show* method

```
In [1]: # Import NumPy and Matplotlib
        %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt

        # Define vector v
        v = np.array([1,1])
```

```
# Plots vector v as blue arrow with red dot at origin (0,0) using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.0, head_width=0.20, head_length=0.25)

# Sets limit for plot for x-axis
plt.xlim(-2,2)

# Set major ticks for x-axis
major_xticks = np.arange(-2, 3)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 2)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 3)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```



1.2 Scaling a Vector using a Scalar

For this part of the lab, we will plot the results of scaling vector $\vec{v}$ by the scalar a. Both scalar a and vector $\vec{v}$ have been defined below.

$ a = 3 $

$$\vec{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

1.2.1 TODO: Multiply Vector by Scalar and Plot Results

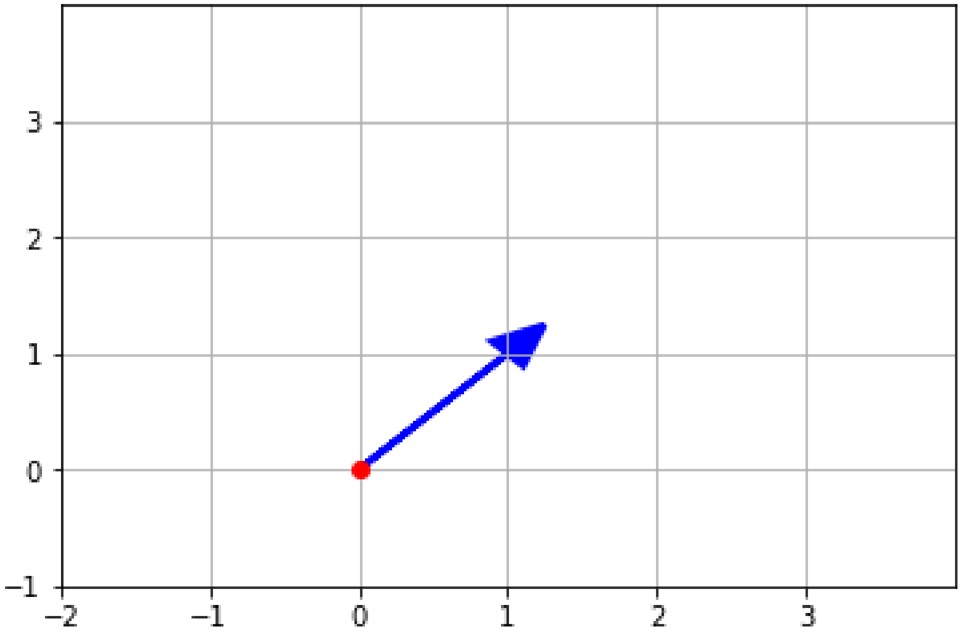
For this part of the lab you will be creating vector $a\vec{v}$ and then adding to the plot as a dotted cyan colored vector.

1. Multiply vector $\vec{v}$ by scalar a in the code below (see ***TODO 1.:***).

2. Use the *ax.arrow(...)* statement in the code below to add vector $a\vec{v}$ to the plot (see **TODO 2.:*). Adding *linestyle = 'dotted'* and changing *color = 'c'* in the *ax.arrow(...)* statement will make vector $a\vec{v}$ a dotted cyan colored vector.

```
In [2]: # Define vector v
        v = np.array([1,1])

        # Define scalar a
        a = 3
```

```
# TODO 1.: Define vector av - as vector v multiplied by scalar a
av = None

# Plots vector v as blue arrow with red dot at origin (0,0) using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# TODO 2.: Plot vector av as dotted (linestyle='dotted') vector of cyan color (color='c'
# using ax.arrow() statement above as template for the plot



# Sets limit for plot for x-axis
plt.xlim(-2, 4)

# Set major ticks for x-axis
major_xticks = np.arange(-2, 4)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```



1.2.2 Solution to Scaling a Vector

Your output from above should match the output below. If you need any help or want to check your answer, feel free to check out the solution notebook by clicking here.

1.2.3 Solution Video for Scaling a Vector

The solution video can be found in the **Vectors Lab Solution** section. You may want to open another browser window to allow you to easily toggle between the Vector's Lab Jupyter Notebook and the solution videos for this lab.

1.3 Adding Two Vectors Together

For this part of the lab, we will plot the result of adding vector $\vec{w}$ to vector $\vec{v}$. Both vectors $\vec{v}$ and $\vec{w}$, have been defined below.

$$\vec{v} = \begin{bmatrix}1\\1\end{bmatrix}$$

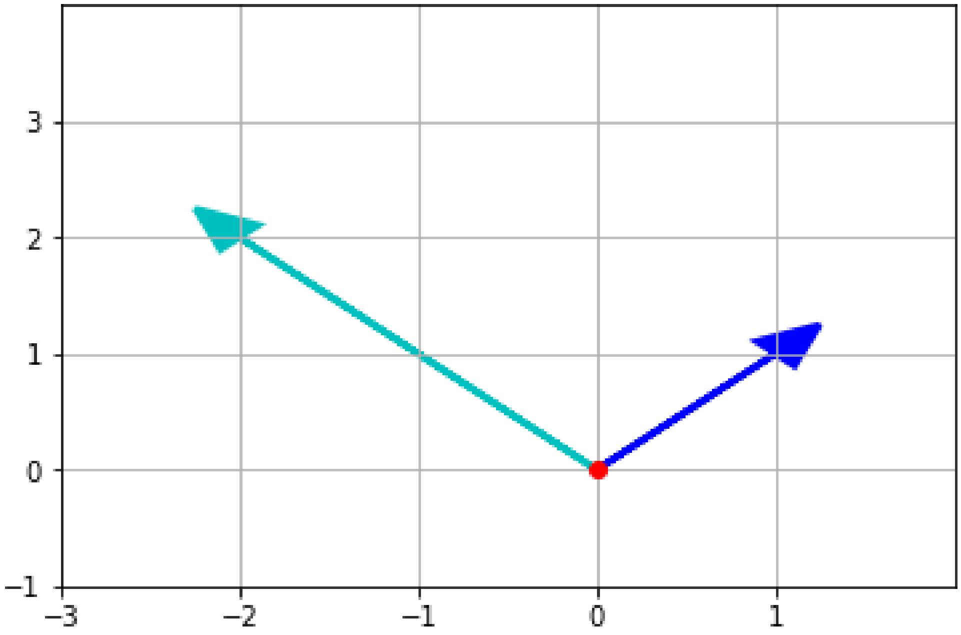
$$\vec{w} = \begin{bmatrix}-2\\2\end{bmatrix}$$

1.3.1 Plotting Two Vectors

The code and the plot that displays vectors $\vec{v}$ and $\vec{w}$ from origin (0,0) can be found below.

```
In [3]: # Define vector v
        v = np.array([1,1])
```

```
# Define vector w
w = np.array([-2,2])

# Plots vector v(blue arrow) and vector w(cyan arrow) with red dot at origin (0,0)
# using Matplotlib

# Creates axes of plot referenced 'ax'
ax = plt.axes()

# Plots red dot at origin (0,0)
ax.plot(0,0,'or')

# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# Plots vector w as cyan arrow starting at origin 0,0
ax.arrow(0, 0, *w, color='c', linewidth=2.5, head_width=0.30, head_length=0.35)

# Sets limit for plot for x-axis
plt.xlim(-3, 2)

# Set major ticks for x-axis
major_xticks = np.arange(-3, 2)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```



1.3.2 Vector Addition

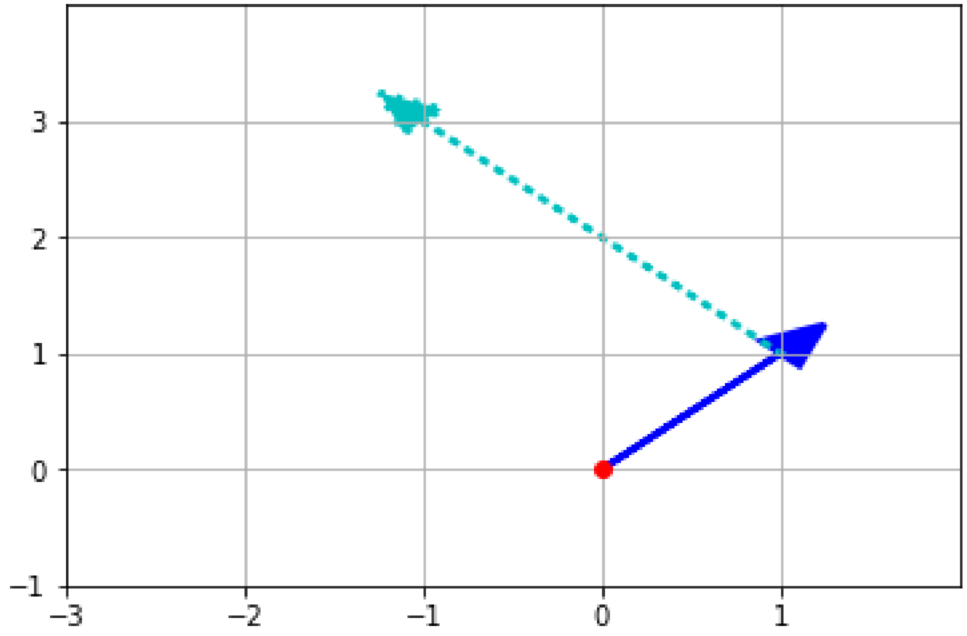
Below we display graphically, adding vector $\vec{w}$ to vector $\vec{v}$.

1.3.3 Plotting Vector Addition

The code and plot that display adding vector $\vec{w}$ to vector $\vec{v}$ can be found below. Notice when we add vector $\vec{w}$ to vector $\vec{v}$, vector $\vec{w}$'s origin is now (1,1). Additionally, we have added *linestyle* = *'dotted'* and changed *color* = *'c'* in the *ax.arrow(...)* statement to make vector $\vec{w}$ a dotted cyan colored vector.

```
In [4]: # Define vector v
        v = np.array([1,1])

        # Define vector w
        w = np.array([-2,2])

        # Plot that graphically shows vector w(dotted cyan arrow) added to vector v(blue arrow)
        # using Matplotlib

        # Creates axes of plot referenced 'ax'
        ax = plt.axes()

        # Plots red dot at origin (0,0)
        ax.plot(0,0,'or')
```

```
# Plots vector v as blue arrow starting at origin 0,0
ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

# Plots vector w as cyan arrow with origin defined by vector v
ax.arrow(v[0], v[1], *w, linestyle='dotted', color='c', linewidth=2.5,
         head_width=0.30, head_length=0.35)

# Sets limit for plot for x-axis
plt.xlim(-3, 2)

# Set major ticks for x-axis
major_xticks = np.arange(-3, 2)
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```



1.3.4 TODO: Adding Two Vectors and Plotting Results

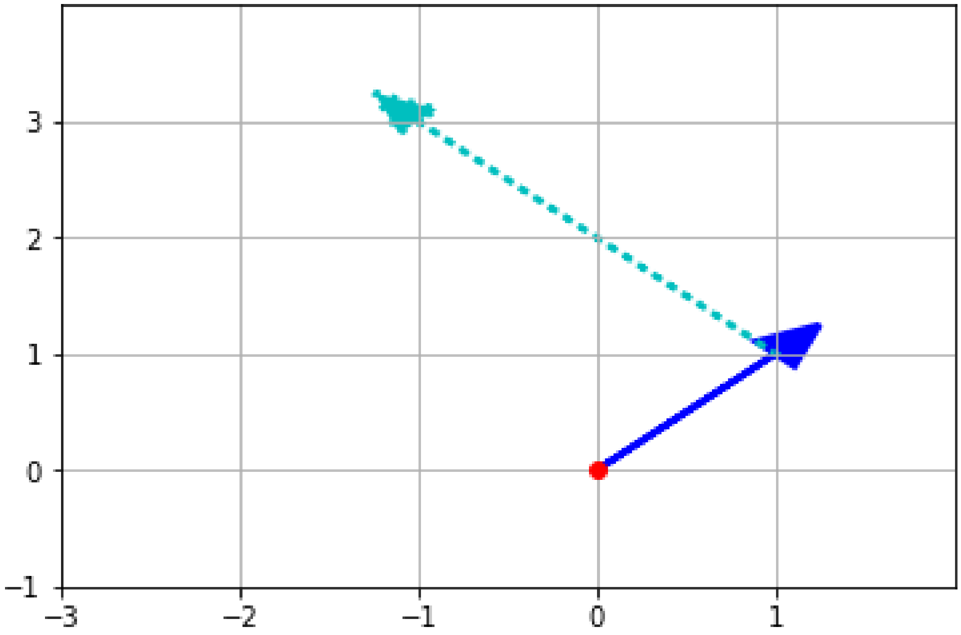
For this part of the lab you will be creating vector $\vec{vw}$ and then adding it to the plot as a thicker width **black** colored vector.

1. Create vector $\vec{vw}$ by adding vector $\vec{w}$ to vector $\vec{v}$ in the code below (see ***TODO 1.:***).

2. Use the *ax.arrow(...)* statement in the code below to add vector $\vec{vw}$ to the plot (see **TODO 2.:*). Changing *linewidth = 3.5* and *color = 'k'* in the *ax.arrow(...)* statement will make vector $\vec{vw}$ a thicker width black colored vector.

```
In [5]: # Define vector v
        v = np.array([1,1])

        # Define vector w
        w = np.array([-2,2])

        # TODO 1.: Define vector vw by adding vectors v and w
        vw = None

        # Plot that graphically shows vector vw (color='b') - which is the result of
        # adding vector w(dotted cyan arrow) to vector v(blue arrow) using Matplotlib

        # Creates axes of plot referenced 'ax'
        ax = plt.axes()

        # Plots red dot at origin (0,0)
        ax.plot(0,0,'or')

        # Plots vector v as blue arrow starting at origin 0,0
        ax.arrow(0, 0, *v, color='b', linewidth=2.5, head_width=0.30, head_length=0.35)

        # Plots vector w as cyan arrow with origin defined by vector v
        ax.arrow(v[0], v[1], *w, linestyle='dotted', color='c', linewidth=2.5,
                 head_width=0.30, head_length=0.35)

        # TODO 2.: Plot vector vw as black arrow (color='k') with 3.5 linewidth (linewidth=3.5)
        # starting vector v's origin (0,0)



        # Sets limit for plot for x-axis
        plt.xlim(-3, 2)

        # Set major ticks for x-axis
        major_xticks = np.arange(-3, 2)
```

```
ax.set_xticks(major_xticks)


# Sets limit for plot for y-axis
plt.ylim(-1, 4)

# Set major ticks for y-axis
major_yticks = np.arange(-1, 4)
ax.set_yticks(major_yticks)

# Creates gridlines for only major tick marks
plt.grid(b=True, which='major')

# Displays final plot
plt.show()
```



1.3.5 Solution to Adding Two Vectors

Your output from above should match the output below. If you need any help or want to check your answer, feel free to check out the solution notebook by clicking here.

1.3.6 Solution Video for Adding Two Vectors

The solution video can be found in the **Vectors Lab Solution** section. You may want to open another browser window to allow you to easily toggle between the Vector's Lab Jupyter Notebook and the solution videos for this lab.