

---

## An Introduction to Vector Operations in Mathematica

In this classnote, we will learn how to do basic vector calculations in Mathematica, and also see how very simple Mathematica programs can be written.

### *Basic Vector Operations :*

We write vectors in Mathematica as a list of components. Consider the vectors (written in Cartesian coordinates as) :

$$\mathbf{v1} = 2 \hat{\mathbf{x}} + 3 \hat{\mathbf{y}} + 2 \hat{\mathbf{z}} \text{ and } \mathbf{v2} = 3 \hat{\mathbf{x}} - 4 \hat{\mathbf{y}} - 3 \hat{\mathbf{z}}$$

We express the vectors in Mathematica as :

```
In[97]:= Clear[v1, v2]
         v1 = {2, 3, 2}
         v2 = {3, -4, -3}
```

Adding the two vectors yields :

```
In[100]:= v1 + v2
Out[100]= {5, -1, -1}
```

(It is always good programming practice to reset, or Clear variables before using them.)

Subtracting vectors gives us :

```
In[101]:= v1 - v2
Out[101]= {-1, 7, 5}
```

We can take the dot and cross products in two different ways :

```
In[102]:= v1.v2
Out[102]= -12
```

Where we use the period key to represent the dot operation, or

```
In[103]:= Dot[v1, v2]
Out[103]= -12
```

The cross product can also be calculated in two ways :

```
In[104]:= v1 × v2
```

```
Out[104]= {-1, 12, -17}
```

where we type `[ESC]cross[ESC]` to produce the  $\times$  symbol; alternately we can use the Cross function :

```
In[105]:= Cross[v1, v2]
```

```
Out[105]= {-1, 12, -17}
```

These results should remind you that the dot product produces a scalar, while the cross product produces a vector.

### *Components and Magnitude of a Vector*

Suppose we want to find the magnitude of vectors  $v_1$  and  $v_2$ ; we know that the magnitude of a vector is the square root of the sum of squares of its components, so the most obvious way to find the length of  $v_1$  is to type in its components and calculate :

```
Clear[magv1]
magv1 = Sqrt[2^2 + 3^2 + 2^2]
```

```
Out[106]=  $\sqrt{17}$ 
```

A more elegant approach (and one that does not require you to type in specific numerical components) involves extracting the  $n$ th element of a list. We can extract these elements (here, the components of the vector) by :

```
In[107]:= Clear[lengthv1]
lengthv1 = Sqrt[v1[[1]]^2 + v1[[2]]^2 + v1[[3]]^2]
```

```
Out[108]=  $\sqrt{17}$ 
```

and obtain the same result.  $v_1[[1]]$  "plucks out" the first component of  $v_1$ ,  $v_1[[2]]$  and  $v_1[[3]]$  similarly extract the y and z components of  $v_1$ . Therefore, we simply find the length of  $v_2$  :

```
In[111]:= Clear[lengthv2]
lengthv2 = Sqrt[v2[[1]]^2 + v2[[2]]^2 + v2[[3]]^2]
```

```
Out[112]=  $\sqrt{34}$ 
```

Actually, we did not need to make use of components at all; Mathematica has a built in function, Norm, that computes the magnitude of vectors, but I did want to show you how to extract the  $n$ th element of a list. To close the loop, we find :

```
In[113]:= Norm[v1]
```

```
Out[113]=  $\sqrt{17}$ 
```

```
In[114]:= Norm[v2]
```

```
Out[114]=  $\sqrt{34}$ 
```

as expected.

### ***Vector Angles***

You recall from analytic geometry that the definition of a dot product yields :

$$\mathbf{v1} \cdot \mathbf{v2} = |\mathbf{v1}| |\mathbf{v2}| \cos\theta$$

We can write a simple program to compute the angle between our two vectors :

```
In[117]:=
  Clear[angle]
  angle = ArcCos[v1.v2 / (Norm[v1] Norm[v2])] * (180 /  $\pi$ ) // N
Out[118]= 119.943
```

The factor of  $\frac{180}{\pi}$  is included to convert the answer to degrees since radians is the default unit for angles in *Mathematica*; the notation //N generates the output as a decimal.

We can check the results of our program by using the built - in Mathematica function :

```
In[119]:= VectorAngle[v1, v2] // N
Out[119]= 2.0934
In[120]:= % (180 /  $\pi$ )
Out[120]= 119.943
```

VectorAngle computes the angle between v1 and v2 in radians; "%" instructs Mathematica to take the most recent output, and in this case, multiply it by  $180/\pi$ .

### ***A Quick Dip in the Graphics Pool***

One of the most powerful features of Mathematica is its excellent graphing package. I will combine some of the features used above to show how we can draw these two 3 D vectors. Please look up Graphics, Graphics3D and Line on the documentation center for more details :

```
In[121]:= Graphics3D[{Line[{{0, 0, 0}, {v1[[1]], v1[[2]], v1[[3]]}}],  
  Line[{{0, 0, 0}, {v2[[1]], v2[[2]], v2[[3]]}}]}
```

Out[121]=

