

## Visualizing vectors in 2D

Based on a Worksheet by Mike May, S.J.- maymk@slu.edu

We have been looking at vectors, how they are added, normalized vectors, and dot products, so it seems worthwhile to look at a Maple visualization of everything we have been doing.

First want to load the appropriate commands. The commands for dot product are in the linalg package. The commands for plotting vectors is in the plottools package. We also need the display command from the plots package. (Restart cleans up in case we were already running a Maple worksheet.)

```
> restart: with(linalg): with(plottools): with(plots):  
Warning, the protected names norm and trace have been redefined and unprotected  
Warning, the names arrow and changecoords have been redefined
```

### Visualizing vectors in $\mathbb{R}^2$ , simple operations:

We want to define some two dimensional vectors. We will use a pair of vectors with numbers filled in so we have a specific example and a pair of vectors with variables so we can see the general case.

```
> v1 := [1,2]; v2 := [3,-4];  
   w1 := [a1, a2]; w2 := [b1, b2];  
  
           v1 := [1, 2]  
           v2 := [3, -4]  
           w1 := [a1, a2]  
           w2 := [b1, b2]
```

To add the vectors we use normal addition. Scalar multiplication also works.

```
> "v1+v2" = v1+v2;  
"w1+w2" = w1+w2;  
"3*v1" = 3*v1;  
"5*w2" = 5*w2;  
  
           "v1+v2" = [4, -2]  
           "w1+w2" = [b1 + a1, b2 + a2]  
           "3*v1" = [3, 6]  
           "5*w2" = [3 b1, 3 b2]
```

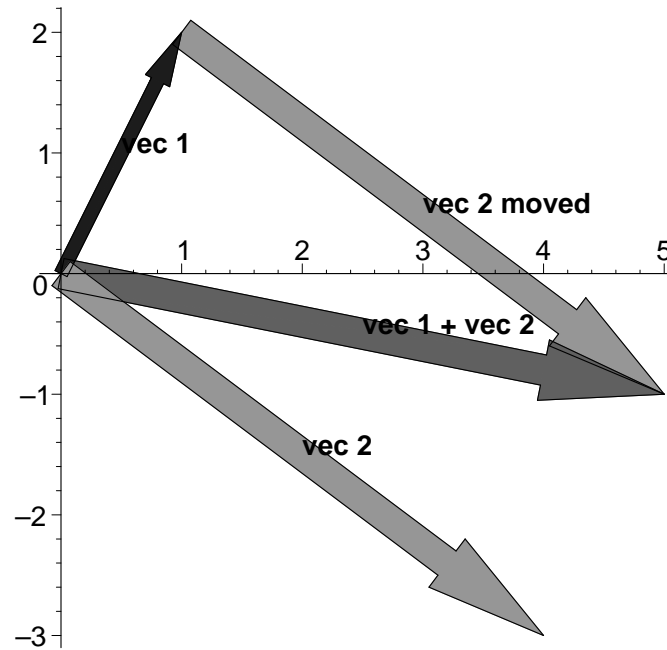
When we try to normalize a vector we have to make a technical note. Like us, Maple has a tendency to leave things like square roots outside the vector. I will use code that forces it to multiply through by the radical, done by the **evalm** command. (You don't need to understand the code. I am only making the comment in case you are interested in why it gets used later on.)

```
> (1/sqrt(5))*v1;  
evalm((1/sqrt(5))*v1);
```

$$\frac{1}{5}\sqrt{5} [1, 2]$$
$$\left[ \frac{\sqrt{5}}{5}, \frac{2\sqrt{5}}{5} \right]$$

Visualizing the vectors and the sum takes a bit more work. The following block of code shows the addition clearly.

```
> #a block of code to look at two vectors and their sum
vec1 := [1,2]:
vec2 := [4, -3]:
vecsum := vec1+vec2;
# Now we are going to plot the vectors in the plane and label
them.
# The following code will
# allow us to put all the vectors on the same plot.
plotv1 := arrow(vec1, color=blue):
label1 := textplot([vec1[1]/2,vec1[2]/2, 'vec 1'],
  align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 10]):
plotv2 := arrow(vec2, color=green):
label2 := textplot([vec2[1]/2,vec2[2]/2, 'vec 2'],
  align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 10]):
plotv2m := arrow(vec1, vec2, color=green):
label2m := textplot([vec1[1]+vec2[1]/2,vec1[2]+vec2[2]/2,
  'vec 2 moved'], align={ABOVE,RIGHT}, font = [HELVETICA,
  BOLD, 10]):
plotvecsum := arrow(vecsum, color=orange):
labelsum := textplot([(vec1[1]+vec2[1])/2,(vec1[2]+vec2[2])/2,
  'vec 1 + vec 2'], align={ABOVE,RIGHT},
  font = [HELVETICA, BOLD, 10]):
display({plotv1, label1, plotv2, label2, plotv2m, label2m,
  plotvecsum, labelsum}, scaling=constrained);
      vecsum := [5, -1]
```



## – Dot products and projections in 2D:

Next we want to look at dot products. First we look at our vectors and unit vectors in the same direction.

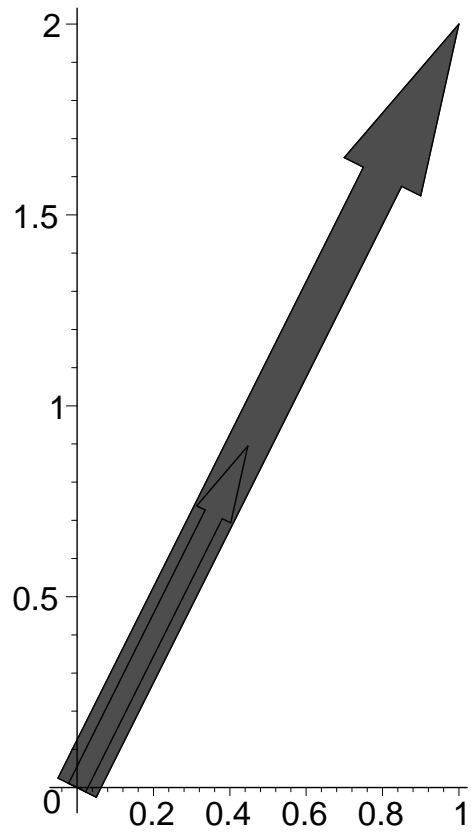
```
> vec1 := v1;
   vec2 := v2;
   magvec1 := sqrt(dotprod(v1,v1));
   vec1norm := evalm(vec1*(1/magvec1));
   magvec2 := sqrt(dotprod(v2,v2));
   vec2norm := evalm(vec2*(1/magvec2));
   plotv1 := arrow(vec1, color=red):
   label1 := textplot([vec1[1]/2,vec1[2]/2, 'vec 1'],
     align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 10]):
   plotv2 := arrow(vec2, color=yellow):
   label2 := textplot([vec2[1]/2,vec2[2]/2, 'vec 2'],
     align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 10]):
   plotv1n := arrow(vec1norm, color=blue):
   plotv2n := arrow(vec2norm, color=brown):
   # A plot showing the vec1 and its unit vector
   display({plotv1,plotv1n},scaling=constrained);
   # A plot showing both of the vectors
   display({plotv1, label1, plotv2, label2, plotv2n, plotv1n},
     scaling=constrained);
```

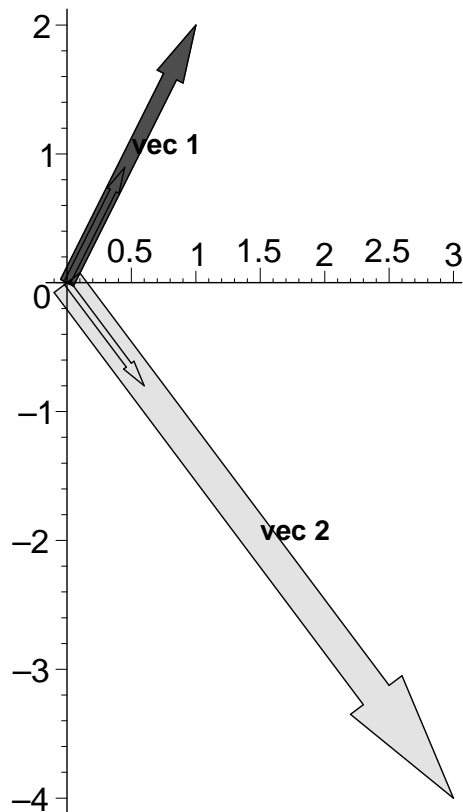
$vec1 := [1, 2]$

$vec2 := [3, -4]$

$$\begin{aligned} \text{magvec1} &:= \sqrt{5} \\ \text{vec1norm} &:= \left[ \frac{\sqrt{5}}{5}, \frac{2\sqrt{5}}{5} \right] \end{aligned}$$

$$\begin{aligned} \text{magvec2} &:= 5 \\ \text{vec2norm} &:= \left[ \frac{3}{5}, \frac{4}{5} \right] \end{aligned}$$





For doing projections, in this class we will always start by normalizing the vector we are projecting onto. (That means we divide it by its length.) That means the projection will be the unit vector times the dot product of the two vectors. Can you identify all the vectors?

```
> #a block of code for finding the projection
#of vec1 onto vec2
vec1 := [3,4];
vec2 := [1,3];
magvec2 := sqrt(dotprod(vec2,vec2));
vec2norm := evalm(vec2*(1/magvec2)); # normalize the vector
we're projecting # onto
dotproduct := dotprod(vec1, vec2norm);
projvec := (evalm(dotproduct*vec2norm)); # do the projection

plotv1 := arrow(vec1,shape=arrow, color=blue):
plotvproj := arrow(projvec,shape=arrow, color=green):
plotv2n := arrow(vec2norm, shape=arrow, color=orange):
plotv2 := arrow(vec2, shape=arrow, color=magenta):
display({plotv1, plotvproj, plotv2n,plotv2},
        scaling=constrained);

vec1 := [3,4]
```

$$\begin{aligned} \text{vec2} &:= [1, 3] \\ \text{magvec2} &:= \sqrt{10} \\ \text{vec2norm} &:= \left[ \frac{\sqrt{10}}{10}, \frac{3\sqrt{10}}{10} \right] \\ \text{dotproduct} &:= \frac{3\sqrt{10}}{2} \\ \text{projvec} &:= \left[ \frac{3}{2}, \frac{9}{2} \right] \end{aligned}$$

