

Visualizing vectors in 2D and 3D

Worksheet by Mike May, S.J.- maymk@slu.edu, ©2006

As we have been looking at vectors, how they are added, normalized vectors, dot and cross products, 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 and cross product are in the LinearAlgebra package. The commands for plotting vectors are 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(LinearAlgebra): with(plottools): with(plots):
```

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

We define some two dimensional vectors. We will use a pair of vectors with numbers 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>;
```

$$v1 := \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$v2 := \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$

$$w1 := \begin{bmatrix} a1 \\ a2 \end{bmatrix}$$

$$w2 := \begin{bmatrix} b1 \\ b2 \end{bmatrix}$$

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;
```

$$"v1+v2" = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$$

$$"w1+w2" = \begin{bmatrix} a1 + b1 \\ a2 + b2 \end{bmatrix}$$

$$"3*v1" = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

$$"5*w2" = \begin{bmatrix} 3 & b1 \\ 3 & b2 \end{bmatrix}$$

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. We will use code that forces it to multiply through by the radical. (You don't need to understand the code. We are only making the comment in case you are interested in why it gets used later on.)

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

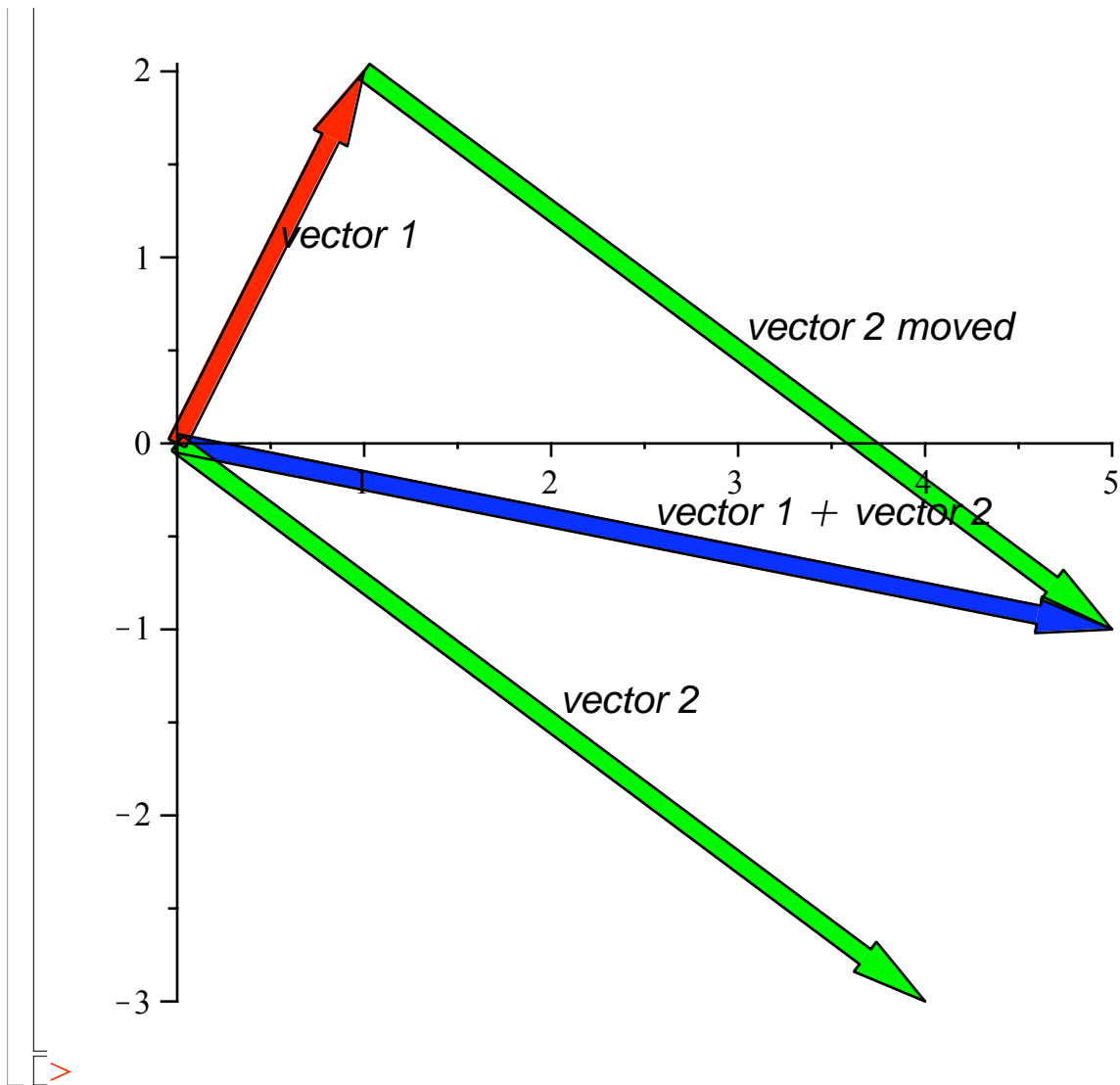
$$\begin{bmatrix} \frac{1}{5} \sqrt{5} \\ \frac{2}{5} \sqrt{5} \end{bmatrix}$$

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

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
zerovec := <0,0>;
vec1 := <1,2>;
vec2 := <4, -3>;
vecsum := vec1+vec2;
plotv1 := arrow(zerovec, vec1, width=.1, head_length=.4,
head_width=.2, color=red):
label1 := textplot([vec1[1]/2,vec1[2]/2, `vector 1`],
align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]):
plotv2 := arrow(zerovec, vec2, width=.1, head_length=.4,
head_width=.2, color=green):
label2 := textplot([vec2[1]/2,vec2[2]/2, `vector 2`],
align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]):
plotv2m := arrow(vec1, vec2, width=.1, head_length=.4,
head_width=.2, color=green):
label2m := textplot([vec1[1]+vec2[1]/2,vec1[2]+vec2[2]/2,
`vector 2 moved`], align={ABOVE,RIGHT}, font =
[HELVETICA, BOLD, 12]):
plotvecsum := arrow(zerovec, vecsum, width=.1, head_length=
.4, head_width=.2, color=blue):
labelsum := textplot([(vec1[1]+vec2[1])/2,(vec1[2]+vec2[2])
/2,
`vector 1 + vector 2`], align={ABOVE,RIGHT},
font = [HELVETICA, BOLD, 12]):
display({plotv1, label1, plotv2, label2, plotv2m, label2m,
plotvecsum, labelsum}, scaling=constrained);
```

$$vecsum := \begin{bmatrix} 5 \\ -1 \end{bmatrix}$$



▼ 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 directions.

```
> zerovec := <0,0>;
vec1 := v1;
vec2 := v2;
magvec1 := sqrt(v1.v1);
vec1norm := vec1*(1/magvec1);
magvec2 := sqrt(v2.v2);
vec2norm := vec2*(1/magvec2);
plotv1 := arrow(zerovec, vec1, width=.1, head_length=.4,
head_width=.2, color=red):
label1 := textplot([vec1[1]/2,vec1[2]/2, `vector 1`],
align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]):
plotv2 := arrow(zerovec, vec2, width=.1, head_length=.4,
head_width=.2, color=yellow):
label2 := textplot([vec2[1]/2,vec2[2]/2, `vector 2`],
align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]):
plotv1n := arrow(zerovec, vec1norm, width=.15, head_length=
.4, head_width=.2, color=blue):
plotv2n := arrow(zerovec, vec2norm, width=.15, head_length=
.4, head_width=.2, color=brown):
```

```
display({plotv1, label1, plotv2, label2, plotv2n, plotv1n},  
        scaling=constrained);
```

$$vec1 := \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

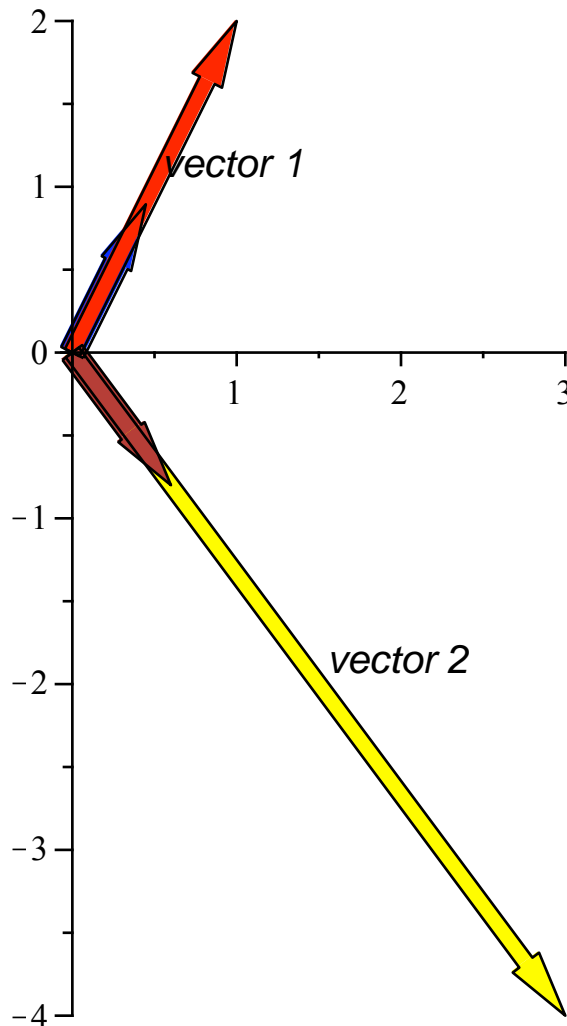
$$vec2 := \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$

$$magvec1 := \sqrt{5}$$

$$vec1norm := \begin{bmatrix} \frac{1}{5} \sqrt{5} \\ \frac{2}{5} \sqrt{5} \end{bmatrix}$$

$$magvec2 := 5$$

$$vec2norm := \begin{bmatrix} \frac{3}{5} \\ -\frac{4}{5} \end{bmatrix}$$



For computing 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.

```
> #a block of code for finding the projection
#of vec1 onto vec2
zerovec := <0,0>;
vec1 := <3,4>;
vec2 := <1,3>;
magvec2 := sqrt(vec2.vec2);
vec2norm := vec2*(1/magvec2);
dotproduct := vec1.vec2norm;
projvec := dotproduct*vec2norm;
plotv1 := arrow(zerovec, vec1, width=.1, head_length=.4,
head_width=.2, color=red);
plotvproj := arrow(zerovec, projvec, width=.1, head_length=
.4, head_width=.14, color=yellow);
plotv2n := arrow(zerovec, vec2norm, width=.1, head_length=.4,
head_width=.22, color=green);
label1 := textplot([vec1[1]/2,vec1[2]/2, `vector 1`],
align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]);
label2 := textplot([vec2norm[1]/2,vec2norm[2]/2, `vector 2
normalized`],
align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]);
labelproj := textplot([projvec[1]/2,projvec[2]/2, `projection
```

```
vector`],  
    align={ABOVE,RIGHT}, font = [HELVETICA, BOLD, 12]):  
display({plotv1, plotvproj, plotv2n, label1, label2,  
labelproj},  
    scaling=constrained);
```

$$vec1 := \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

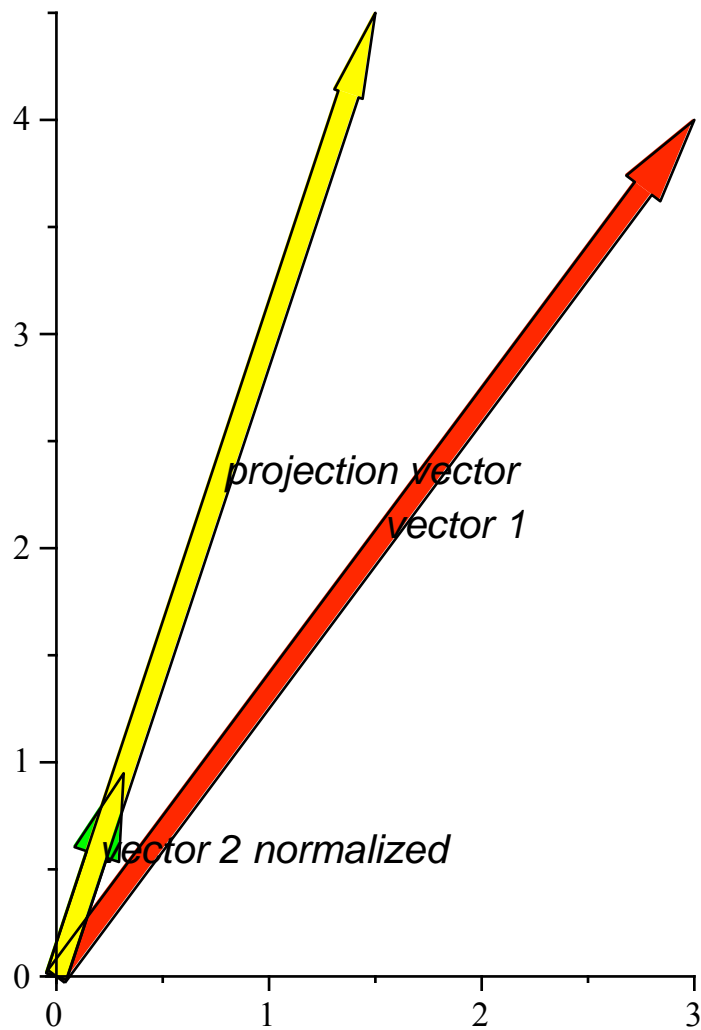
$$vec2 := \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$magvec2 := \sqrt{10}$$

$$vec2norm := \begin{bmatrix} \frac{1}{10} \sqrt{10} \\ \frac{3}{10} \sqrt{10} \end{bmatrix}$$

$$dotproduct := \frac{3}{2} \sqrt{10}$$

$$projvec := \begin{bmatrix} \frac{3}{2} \\ \frac{9}{2} \end{bmatrix}$$



Visualizing in 3D

We now want to repeat everything in 3 dimensions. We start by defining some vectors.

```
> v3 := <1, 5, 3>; v4 := <4, 2, 6>;
   w3 := <c1, c2, c3>; w4 := <d1, d2, d3>;
```

$$v3 := \begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix}$$

$$v4 := \begin{bmatrix} 4 \\ 2 \\ 6 \end{bmatrix}$$

$$w3 := \begin{bmatrix} c1 \\ c2 \\ c3 \end{bmatrix}$$

$$w4 := \begin{bmatrix} d1 \\ d2 \\ d3 \end{bmatrix}$$

Addition of vectors and scalar multiplication works the same way it did in 2D.

```
> "v3+v4 =" = v3+v4;
" w3+w4 =" = w3+w4;
" 3*v3 =" = 3*v1;
" 5*w4 =" = 3*w4;
```

$$"v3+v4 =" = \begin{bmatrix} 5 \\ 7 \\ 9 \end{bmatrix}$$

$$"w3+w4 =" = \begin{bmatrix} c1 + d1 \\ c2 + d2 \\ c3 + d3 \end{bmatrix}$$

$$"3*v3 =" = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

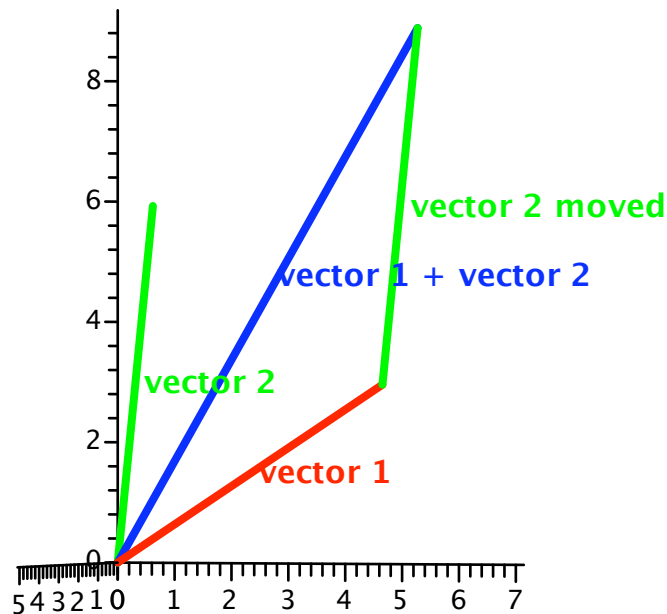
$$"5*w4 =" = \begin{bmatrix} 3 d1 \\ 3 d2 \\ 3 d3 \end{bmatrix}$$

Plotting once again shows that vector addition is done by moving vectors.

```
>
> #a block of code for adding 2 vectors in 3D.
zerovec := <0,0,0>;
vec1 := v3:
vec2 := v4:
normvec := <-1,-1,1>:
vecsum := vec1+vec2:
plotv1 := line(convert(zerovec,list), convert(vec1,list),
normvec, linestyle=1,
thickness=3, color=red):
label1 := textplot3d([vec1[1]/2,vec1[2]/2,vec1[3]/2, ` vector
1`],
align=RIGHT, font = [HELVETICA, BOLD, 12], color=red):
plotv2 := line(convert(zerovec,list), convert(vec2,list),
normvec, linestyle=1,
thickness=3, color=green):
label2 := textplot3d([vec2[1]/2,vec2[2]/2,vec2[3]/2, ` vector
2`],
align=RIGHT, font = [HELVETICA, BOLD, 12],color=green):
plotv2m := line(convert(vec1,list), convert(vecsum,list),
normvec, linestyle=1,
thickness=3, color=green):
label2m := textplot3d([vec1[1]+vec2[1]/2, vec1[2]+vec2[2]/2,
vec1[3]+vec2[3]/2, ` vector 2 moved`], align=RIGHT,
font = [HELVETICA, BOLD, 12], color=green):
plotvecsum := line(convert(zerovec,list), convert(vecsum,
```

```
list), normvec, linestyle=1,
    thickness=3, color=blue):
labelsum := textplot3d([(vec1[1]+vec2[1])/2, (vec1[2]+vec2[2]
)/2,
    (vec1[3]+vec2[3])/2, `vector 1 + vector 2`], align=
{ABOVE,RIGHT},
    font = [HELVETICA, BOLD, 12], color=blue):
display({plotv1, label1, plotv2, label2, plotv2m, label2m,
    labelsum, plotvecsum}, axes=normal, scaling=constrained);
```

$$\text{zerovec} := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



>

▼ Dot products and projections in 3D:

We again repeat everything we did in two dimensions, but with three dimensions. First we look at our vectors and unit vectors in the same directions.

```
> zerovec := <0,0, 0>;
vec1 := v3;
vec2 := v4;
magvec1 := sqrt(vec1.vec1);
```

```

vec1norm := vec1*(1/magvec1);
magvec2 := sqrt(vec2.vec2);
vec2norm := vec2*(1/magvec2);
plotv1 := line(convert(zerovec,list), convert(vec1,list),
linestyle=1,
thickness=1, color=red):
label1 := textplot3d([vec1[1]/2,vec1[2]/2,vec1[3]/2, ` vector
1`],
align=RIGHT, font = [HELVETICA, BOLD, 12], color=red):
plotv2 := line(convert(zerovec,list), convert(vec2,list),
linestyle=1,
thickness=1, color=green):
label2 := textplot3d([vec2[1]/2,vec2[2]/2,vec2[3]/2, ` vector
2`],
align=RIGHT, font = [HELVETICA, BOLD, 12],color=green):
plotv1n := line(convert(zerovec,list), convert(vec1norm,list)
,linestyle=1,
thickness=3, color=blue):
plotv2n := line(convert(zerovec,list), convert(vec2norm,list)
, linestyle=1,
thickness=3, color=brown):
display3d({plotv1, label1, plotv2, label2,
plotv2n, plotv1n},axes=normal,scaling=constrained);

```

$$vec1 := \begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix}$$

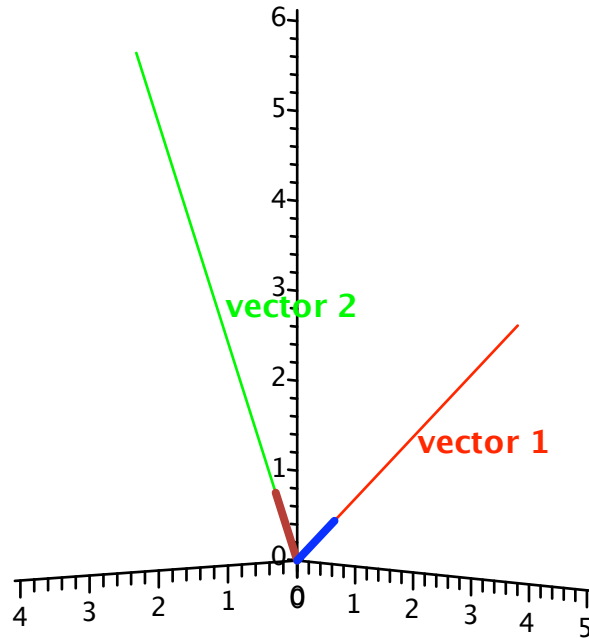
$$vec2 := \begin{bmatrix} 4 \\ 2 \\ 6 \end{bmatrix}$$

$$magvec1 := \sqrt{35}$$

$$vec1norm := \begin{bmatrix} \frac{1}{35} \sqrt{35} \\ \frac{1}{7} \sqrt{35} \\ \frac{3}{35} \sqrt{35} \end{bmatrix}$$

$$magvec2 := 2\sqrt{14}$$

$$vec2norm := \begin{bmatrix} \frac{1}{7} \sqrt{14} \\ \frac{1}{14} \sqrt{14} \\ \frac{3}{14} \sqrt{14} \end{bmatrix}$$



Repeating the process we used in 2D, we normalize before projecting.

```
> zerovec := <0,0, 0>;
vec1 := v3;
vec2 := v4;
magvec2 := sqrt(vec2.vec2);
vec2norm := vec2*(1/magvec2);
dotproduct := vec1.vec2norm;
projvec := dotproduct*vec2norm;
magvec2 := sqrt(vec2.vec2);
plotv1 := line(convert(zerovec,list), convert(vec1,list),
linestyle=1,
thickness=1, color=red):
labell1 := textplot3d([vec1[1]/2,vec1[2]/2,vec1[3]/2, ` vector
1`],
align=RIGHT, font = [HELVETICA, BOLD, 12], color=red):
plotv2n := line(convert(zerovec,list), convert(vec2norm,list)
, linestyle=1,
thickness=3, color=brown):
label2n := textplot3d([vec2norm[1]/2,vec2norm[2]/2,vec2norm
[3]/2,
` vector 2 normalized`], align=RIGHT,
font = [HELVETICA, BOLD, 12], color=brown):
plotvperp := line(convert(zerovec,list), convert(projvec,
list), linestyle=1,
thickness=1, color=black):
```

```

plotvproj := line(convert(zerovec,list), convert(projvec,
list), linestyle=1,
thickness=1, color=green):
labelproj := textplot3d([projvec[1]/2,projvec[2]/2,projvec[3]
/2,
` projection vector`],
align=RIGHT, font = [HELVETICA, BOLD, 12],color=green):
display3d({plotv1, labell1, plotv2n, label2n, plotvproj,
labelproj,
plotvperp}, axes=normal,scaling=constrained);

```

$$vec1 := \begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix}$$

$$vec2 := \begin{bmatrix} 4 \\ 2 \\ 6 \end{bmatrix}$$

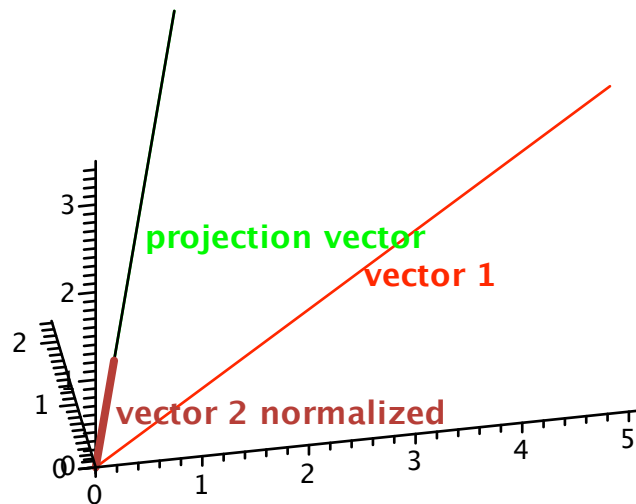
$$magvec2 := 2\sqrt{14}$$

$$vec2norm := \begin{bmatrix} \frac{1}{7}\sqrt{14} \\ \frac{1}{14}\sqrt{14} \\ \frac{3}{14}\sqrt{14} \end{bmatrix}$$

$$dotproduct := \frac{8}{7}\sqrt{14}$$

$$projvec := \begin{bmatrix} \frac{16}{7} \\ \frac{8}{7} \\ \frac{24}{7} \end{bmatrix}$$

$$magvec2 := 2\sqrt{14}$$



▼ The cross product of vectors in 3D:

The syntax for cross product is simple.

```
> v3 := <-1, 3, -2>; v4 := <3, -2, 1>;
   "v3 x v4 =" = CrossProduct (v3, v4);
```

$$v3 := \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix}$$

$$v4 := \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$$

$$"v3 \times v4 =" = \begin{bmatrix} -1 \\ -5 \\ -7 \end{bmatrix}$$

We want to observe that the cross product is perpendicular to both of the other vectors.

```

> zerovec := <0,0, 0>;
vec1 := v3;
vec2 := v4;
cornvec:= vec1+vec2;
crossvec := CrossProduct(vec1, vec2);
plotv1 := line(convert(zerovec,list), convert(vec1,list),
linestyle=1,
    thickness=1, color=red);
label1 := textplot3d([vec1[1]/2,vec1[2]/2,vec1[3]/2, ` vector
1`],
    align=RIGHT, font = [HELVETICA, BOLD, 12], color=red);
plotv2 := line(convert(zerovec,list), convert(vec2,list),
linestyle=1,
    thickness=3, color=brown);
label2 := textplot3d([vec2[1]/2,vec2[2]/2,vec2[3]/2, ` vector
2`],
    align=RIGHT, font = [HELVETICA, BOLD, 12], color=brown);
plotvcross := line(convert(zerovec,list), convert(crossvec,
list), linestyle=1,
    thickness=3, color=blue);
labelcross := textplot3d([crossvec[1]/2,crossvec[2]/2,
crossvec[3]/2,
    ` vector 1 cross vector 2`],
    align=RIGHT, font = [HELVETICA, BOLD, 12], color=blue);
plotpoly := polygon([convert(zerovec,list), convert(vec1,
list),
    convert(cornvec,list), convert(vec2,
list)],
    thickness=1, color=green);
labelproj := textplot3d([projvec[1]/2,projvec[2]/2,projvec[3]
/2,
    ` projection vector`],
    align=RIGHT, font = [HELVETICA, BOLD, 12],color=green);
display3d({plotv1, label1, plotv2, label2, plotvcross,
labelcross, plotpoly}, axes=normal,scaling=constrained);

```

$$vec1 := \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix}$$

$$vec2 := \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$$

$$crossvec := \begin{bmatrix} -1 \\ -5 \\ -7 \end{bmatrix}$$

