

Homework 2, 6/28/05

At the end of yesterday's session the question came up of how to add a normal line to a plane.

We start with our normal set-up

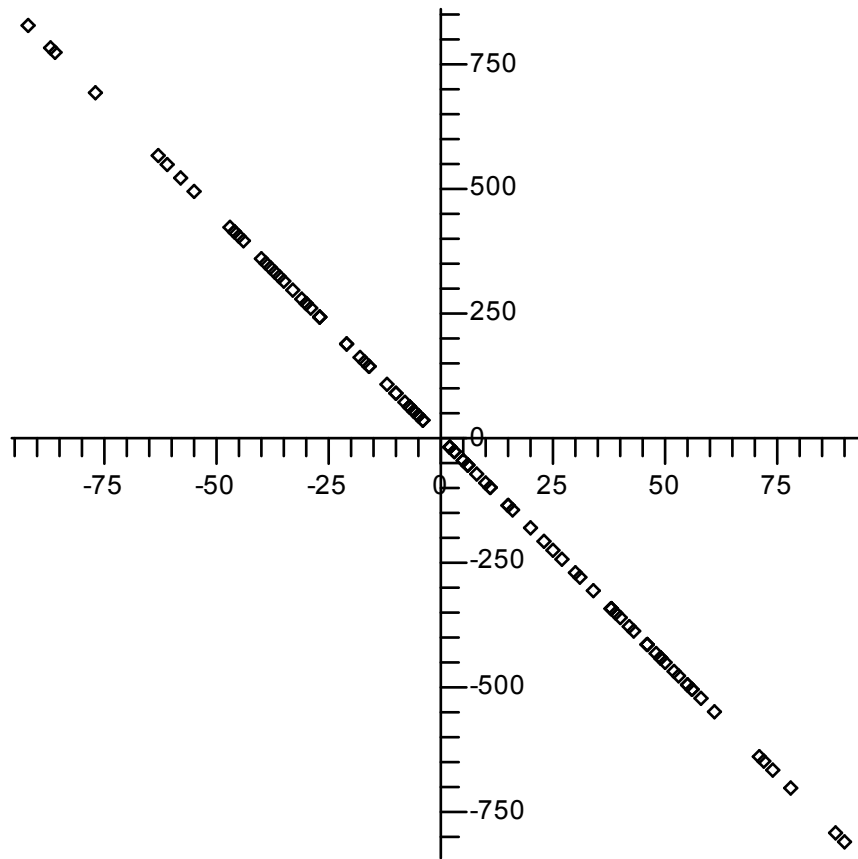
```
> restart: with(LinearAlgebra): with(plots): with(plottools):  
Warning, the name changecoords has been redefined  
Warning, the assigned name arrow now has a global binding
```

We then find the matrix and the set of point like last time.

```
> A := RandomMatrix(1,3,generator=rand(-10..10));  
A := <A,rand(-10..10)()*Row(A,1)>;  
setofpoints :=  
{seq(Vector(3,[rand(-10..10)(),rand(-10..10)(),rand(-10..10)()]),  
i=1..100)}:  
pointplot3d(setofpoints,view=[-10..10,-10..10,-10..10],  
axes=normal,color=black):  
A := [4 2 -5]  
A :=  $\begin{bmatrix} 4 & 2 & -5 \\ -36 & -18 & 45 \end{bmatrix}$  (1)
```

Last time we plotted the image of our 100 random points and found a basis of the null space.

```
> setofimages := {seq(A.setofpoints[i],i=1..100)} minus {0}:  
pointplot(setofimages,color=black);  
nullbasis:=NullSpace(A);
```

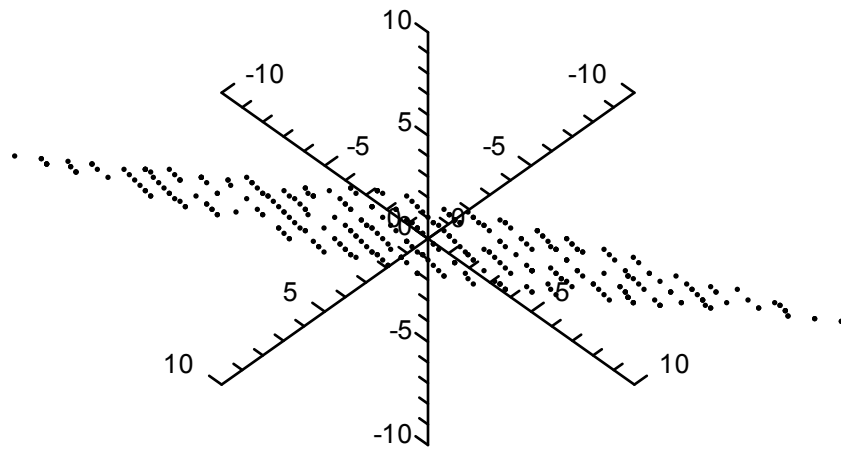


$$\text{nullbasis} := \left\{ \begin{bmatrix} \frac{5}{4} \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{bmatrix} \right\}$$

(2)

We then plotted linear combinations of the basis vectors of the null space.

```
> nullpoints := {seq(rand(-10..10)*nullbasis[1] +
    rand(-10..10)*nullbasis[2], i=1..500)} minus {0}:
pointplot3d(nullpoints, view=[-10..10, -10..10, -10..10],
    axes=normal, color=black);
```



We want to take a cross product to get a normal vector.

```
> v1 := nullbasis[1]; v2 := nullbasis[2]; v3 := CrossProduct(v1,v2);
```

$$v1 := \begin{bmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{bmatrix}$$

$$v2 := \begin{bmatrix} \frac{5}{4} \\ 0 \\ 1 \end{bmatrix}$$

$$v3 := \begin{bmatrix} 1 \\ \frac{1}{2} \\ -\frac{5}{4} \end{bmatrix} \quad (3)$$

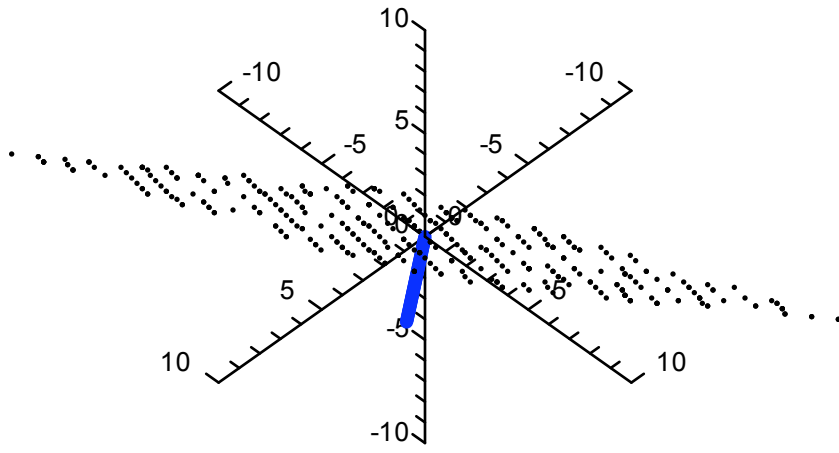
Normalize the normal vector.

```
> v3 := v3/sqrt(v3.v3);
```

$$V3 := \begin{bmatrix} \frac{4}{15} \sqrt{5} \\ \frac{2}{15} \sqrt{5} \\ -\frac{1}{3} \sqrt{5} \end{bmatrix} \quad (4)$$

Now we make the plots of the points and the vector into separate plot structure and display them together.

```
> SC := spacecurve(V3*t, t=0..3, thickness = 5,color=blue):
PP := pointplot3d(nullpoints,view=[-10..10,-10..10,-10..10],
axes=normal,color=black):
display3d(SC,PP);
```



[>