Image of a Linear Transformation

Worksheet by Russell Blyth

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

Outline

The basic objectives are:

1) Investigate the image of a particular linear transformation.
2) Graphically investigate the null space of the linear transformation.

```
>
```

Working with the image of a linear transformation

Define a random 2 x 3 matrix of rank 1.

```
> A := RandomMatrix(1,3,generator=rand(-10..10));
```

$$A := \begin{bmatrix} 4 & 2 & -5 \end{bmatrix} \tag{2.1}$$

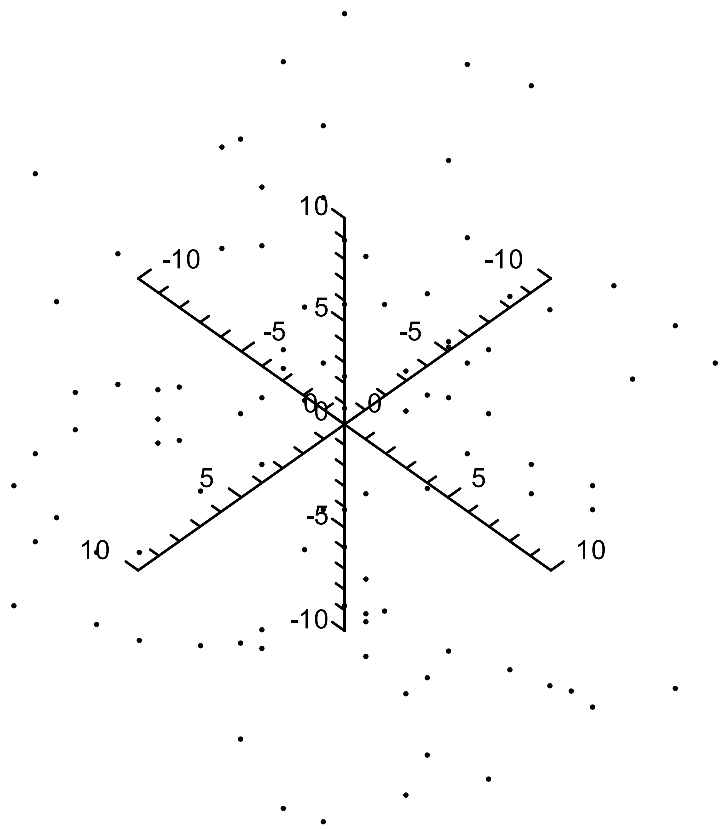
```
> S:=rand(-10..10)():
> A := < A, S * Row(A, 1)>;
```

$$A := \begin{bmatrix} 4 & 2 & -5 \\ -36 & -18 & 45 \end{bmatrix} \tag{2.2}$$

Interpret A as a matrix that represents a linear transformation T from R^3 to R^2

What is the dimension of the image of T? We investigate by creating 100 random points in R^3 and finding and plotting the images of these 100 points under multiplication on the left by the matrix A. First the 100 points:

```
> 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);
```

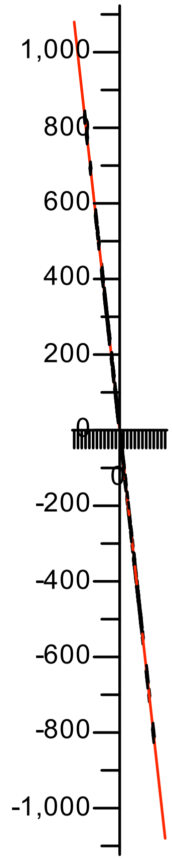


Rotate the plot to see that the points are spread around in R^3.

Next, plot the images of these points:

```
> setofimages := {seq(A.setofpoints[i],i=1..100)} minus {0}:
  imagepoints:=pointplot(setofimages,color=black):
> imageline:=plot([t,S*t,t=-120..120],color=red):
```

```
> display3d( {imagepoints, imageline}, scaling = constrained);
```



Questions: Why do the points only cover part of the line? What would you need to change to get points on the uncovered portion of the line? What is the slope of the line (note that the line does not have the slope it appears to have due to the scaling of the axes).

How many of these points are mapped to zero, i.e. in the null space? In defining the set, "setofimages", the points in the null space were excluded. So lets count the number of points left in "setofimages".

```
> nops(setofimages);
```

$$100 \qquad (2.3)$$

Notice that none of our image points were in the null space. To be in the null space a point, X, must satisfy the following matrix equation AX = 0. Or in our case

```
> eq1:=Row(A,1).Vector(<x,y,z>)=0;
```

$$eq1 := 4\,x + 2\,y - 5\,z = 0 \qquad (2.4)$$

```
> eq2 := solve(eq1, x);
```

(2.5)

$$eq2 := -\frac{1}{2}y + \frac{5}{4}z \tag{2.5}$$

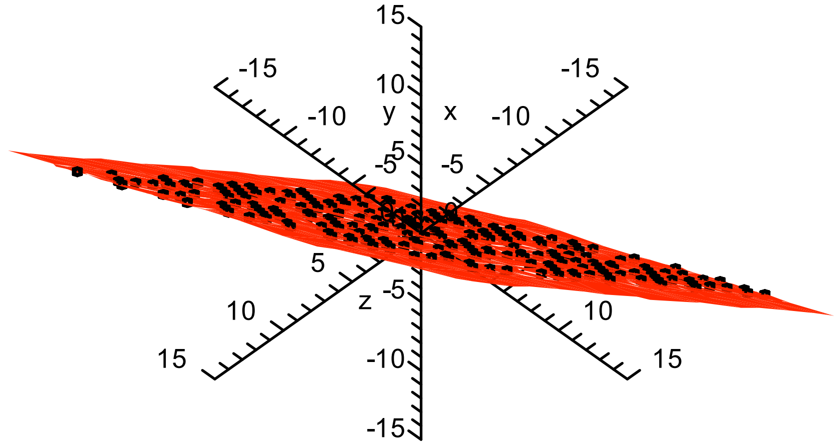
Why were none of our points in the null space? Remember that x, y and z were all defined as integer values between -10 and 10.

Lets plot some points in the nullspace by computing some random values of y and z and use these to find the appropriate value of x.

```
>
> nullpoints:=NULL:

> for i from 1 to 300 do:
  valueofy := rand(-10..10)():
  valueofz := rand(-10..10)():
  nullpoints := nullpoints union {Vector(3,[eval(eq2,[y=valueofy,
  z=valueofz]),valueofy,valueofz])}:
  end do:

>
>
> plotofpoints:=pointplot3d(nullpoints,view=[-15..15,-15..15,-15.
  .15],symbol=circle,color=black):
  plotofplane:=implicitplot3d(eq1,x=-15..15,y=-15..15,z=-15..15,
  color=red,style=patchnogrid,axes=normal,transparency=.5):
  display3d(plotofpoints,plotofplane);
```



>

>

Rotate the plot to see that the null space is a plane, and hence has dimension 2.

Exercise:

1) Create a random 3x3 matrix B of rank 2. Repeat the calculations and plots performed above for the matrix A for your matrix B. Note that B represents a linear transformation from R^3 to R^3, so all plots will be 3D plots.

>

>