

# Image of a Linear Transformation

Worksheet by Russell Blyth.

Edited by Harry S. (Steve) Mills in a feeble effort to assist Regina Souza and Hugh Sanders put something together to help students visualize linear transformations. My particular part was to try and color-code points in the domain and tie them to their respective partners in the image under the mapping whose matrix is  $A$ , reasoning that the color-coding would be a nice touch. It really isn't, in my opinion. What follows in this worksheet is some play involving a linear transformation of rank 1 from  $\mathbb{R}^3$  into  $\mathbb{R}^2$ .

Unlike the (wonderful!) car example put together by our (fearless) leaders, the point, here, is to hammer home the "Image is a subspace" idea. The color coding idea (mine) was so that students could play around and see what's being sent where. I'm still playing around with symbol sizes, the optimal # of points to use for student understanding.

Overall, I'm not particularly happy with what I'm getting. And this "brute force" method has obvious disadvantages. But you may see something here that does you some good, with respect to plotting. Ultimately, I'd like to build a procedure that ties the indexing variable to the colors used for image and pre-image. Then I can (hopefully) use Regina's idea to put image and pre-image side-by-side in a table, using document mode.

I think that to do this, I'll want to save the procedure locally and load it into the document, to spare the student the coding overhead.

Both these worksheets are under construction, and meant to supplement the fine work already done by Hugh Sanders and Regina Souza. I'll add it to what they've done (if that's called for) when I've cleaned it up a bit.

The worksheet "Mills Homework 3D.mw" is a rank 2 mapping on  $\mathbb{R}^3$ . Again, color-izing the random points doesn't seem to add much. The fact that the "cloud" is being sent to a plane thru the origin is the point and is not really helped by colors.

As a student, I feel like a failure, here, but as a scientist, sharing a blind alley with my colleagues isn't such a bad thing.

Bottom line: The pedagogy doesn't seem to be helped much by my idea. But I DID gain some insight into how to manipulate images for various purposes.

## ▼ Obligatory Preliminaries

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

## Brute Force use of "display" command.

All I'm doing is hacking around trying to find nice ways of rendering the cloud.

Define a random 2 x 3 matrix of rank 1.

```
> A := RandomMatrix(1,3,generator=rand(-5..5));  
A := <A,rand(-2..2)()*Row(A,1)>;
```

$$A := \begin{bmatrix} 1 & 4 & 0 \\ 1 & 4 & 0 \\ -1 & -4 & 0 \end{bmatrix} \quad (2.1)$$

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:

```
> oct01points :=  
{seq(Vector(3, [rand(0..10)(),rand(0..10)(),rand(0..10)()]),i=1..5)}:  
plot01:=pointplot3d(oct01points,view=[-10..10,-10..10,-10..10],  
axes=normal,symbol=circle,symbolsize=50,color=black):  
images01:={seq(A.oct01points[i],i=1..5)} minus {0}:  
images01plot:=pointplot(images01,symbolsize=50,color=black):  
  
oct02points :=  
{seq(Vector(3, [rand(-10..0)(),rand(0..10)(),rand(0..10)()]),i=1..5)}:  
plot02:=pointplot3d(oct02points,view=[-10..10,-10..10,-10..10],  
axes=normal,symbol=circle,symbolsize=50,color=blue):  
images02:={seq(A.oct02points[i],i=1..5)} minus {0}:  
images02plot:=pointplot(images02,symbolsize=50,color=blue):  
oct03points :=  
{seq(Vector(3, [rand(-10..0)(),rand(-10..0)(),rand(0..10)()]),i=1..5)}:  
plot03:=pointplot3d(oct03points,view=[-10..10,-10..10,-10..10],  
axes=normal,symbol=circle,symbolsize=50,color=red):  
images03:={seq(A.oct03points[i],i=1..5)} minus {0}:  
images03plot:=pointplot(images03,symbolsize=50,color=red):  
oct04points :=  
{seq(Vector(3, [rand(-10..0)(),rand(0..10)(),rand(-10..0)()]),i=1..5)}:  
plot04:=pointplot3d(oct04points,view=[-10..10,-10..10,-10..10],  
axes=normal,symbol=circle,symbolsize=50,color=green):  
images04:={seq(A.oct04points[i],i=1..5)} minus {0}:  
images04plot:=pointplot(images04,symbolsize=50,color=green):  
oct05points :=  
{seq(Vector(3, [rand(-10..0)(),rand(-10..0)(),rand(-10..0)()]),i=1..5)}:  
plot05:=pointplot3d(oct05points,view=[-10..10,-10..10,-10..10],  
axes=normal,symbol=diamond,symbolsize=50,color=cyan):  
images05:={seq(A.oct05points[i],i=1..5)} minus {0}:  
images05plot:=pointplot(images05,symbolsize=50,color=cyan):  
oct06points :=  
{seq(Vector(3, [rand(0..10)(),rand(-10..0)(),rand(0..10)()]),i=1..5)}:  
plot06:=pointplot3d(oct06points,view=[-10..10,-10..10,-10..10],  
axes=normal,symbol=diamond,symbolsize=50,color=magenta):
```

```

images06:={seq(A.oct06points[i],i=1..5)} minus {0}:
images06plot:=pointplot(images06,symbolsize=50,color=magenta):

oct07points :=
{seq(Vector(3,[rand(0..10)(),rand(-10..0)(),rand(-10..0)()]),i=
1..5)}:
plot07:=pointplot3d(oct07points,view=[-10..10,-10..10,-10..10],
axes=normal,symbol=diamond,symbolsize=50,color=gold):

images07:={seq(A.oct07points[i],i=1..5)} minus {0}:
images07plot:=pointplot(images07,symbolsize=50,color=magenta):

oct08points :=
{seq(Vector(3,[rand(0..10)(),rand(0..10)(),rand(-10..0)()]),i=
1..5)}:
plot08:=pointplot3d(oct08points,view=[-10..10,-10..10,-10..10],
axes=normal,symbol=diamond,symbolsize=50,color=gray):

images08:={seq(A.oct08points[i],i=1..5)} minus {0}:
images08plot:=pointplot(images08,symbolsize=50,color=magenta):
display([plot01,plot02,plot03,plot04,plot05,plot06,plot07,
plot08]);
display([images01plot,images02plot,images03plot,images04plot,
images06plot,images06plot,images07plot,images08plot]);

```



