

Visualizing Linear Transformations in \mathbb{R}^3

Worksheet by Michael K. May, S. J., revised 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) Learn how to have Maple draw a stick figure of a car in \mathbb{R}^3 .
- 2) Learn to represent a linear transformation in \mathbb{R}^3 as multiplication by a matrix.
- 3) Explore the relationships between the matrix of a transformation and its image and nullspace.

```
>
```

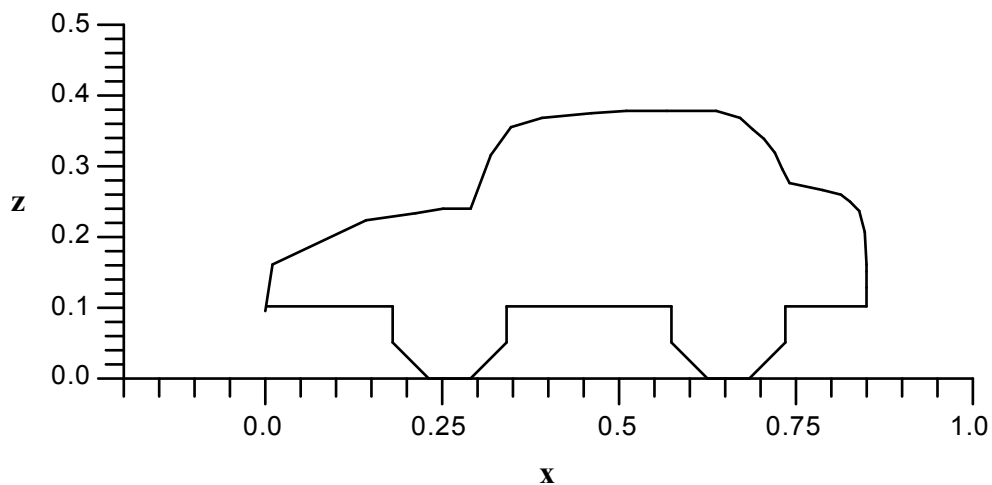
Drawing a 3 dimensional car in Maple

We first need to get Maple to draw a 3 dimensional object. We follow the OnLine section in the student resource manual and work with the image of a car. We start by producing a two dimensional outline of a car.

```
> listcar := [<0, .0954>, <.0102, .1612>, <.1425, .2237>, <.2124,  
.2336>,  
          <.2513, .2401>, <.2902, .2401>, <.3187, .3158>, <.3472,  
.3553>,  
          <.3912, .3684>, <.4611, .3750>, <.5104, .3783>, <.5674,  
.3783>,  
          <.6373, .3783>, <.6710, .3684>, <.6891, .3520>, <.7047,  
.3388>,  
          <.7202, .3191>, <.7306, .2961>, <.7409, .2763>, <.7876,  
.2664>,  
          <.8135, .2599>, <.8264, .2500>, <.8394, .2368>, <.8472,  
.2072>,  
          <.8497, .1612>, <.8497, .1513>, <.8497, .1283>, <.8497,  
.1020>,  
          <.7350, .1020>, <.7350, .0510>, <.6840, .0000>, <.6250,  
.0000>,  
          <.5740, .0510>, <.5740, .1020>, <.3410, .1020>, <.3410,  
.0510>,  
          <.2900, .0000>, <.2310, .0000>, <.1800, .0510>, <.1800,  
.1020>,  
          <.0020, .1020>]:
```

We trace the outline.

```
> pointplot(listcar, style=line, view=[-0.2..1, 0..0.5], axes=  
framed, scaling=constrained, color=black, labels=[x,z],  
labelfont=[TIMES, BOLD, 12], connect=true);
```

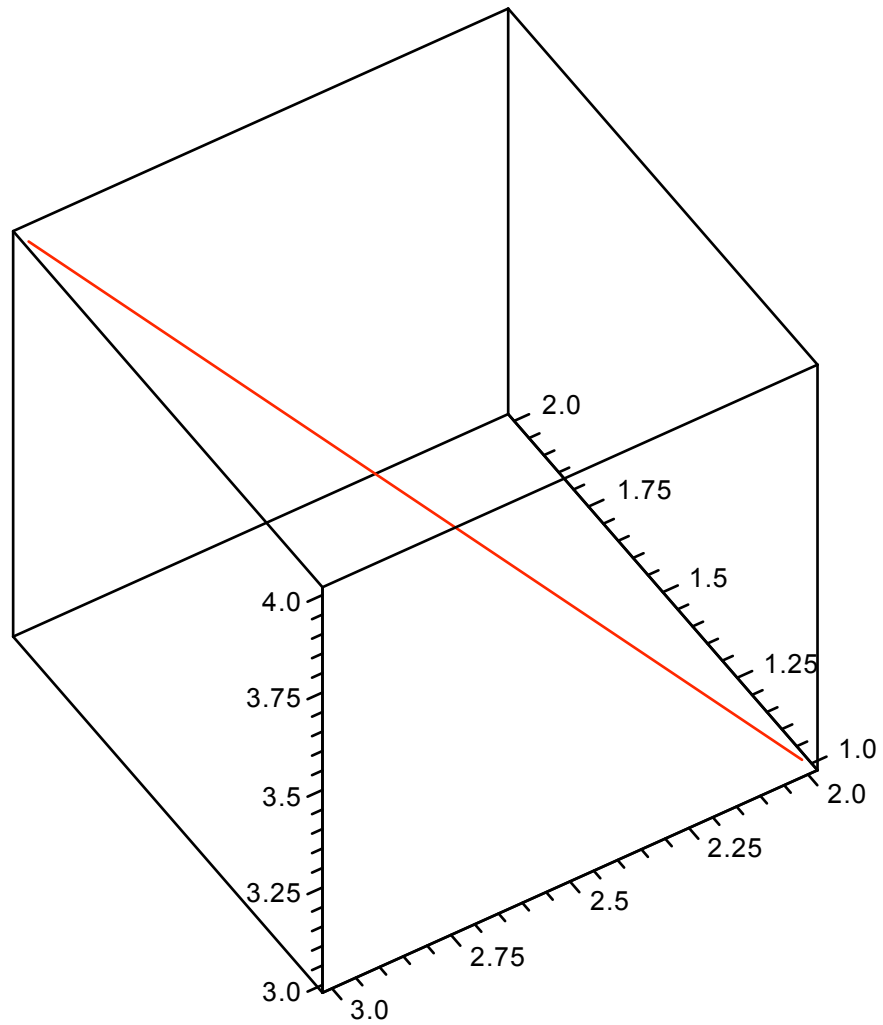


To turn the car into a 3 dimensional object, we insert a y coordinate in between the x and z coordinates. We make the car .35 units wide, with the right side in the x-z plane.

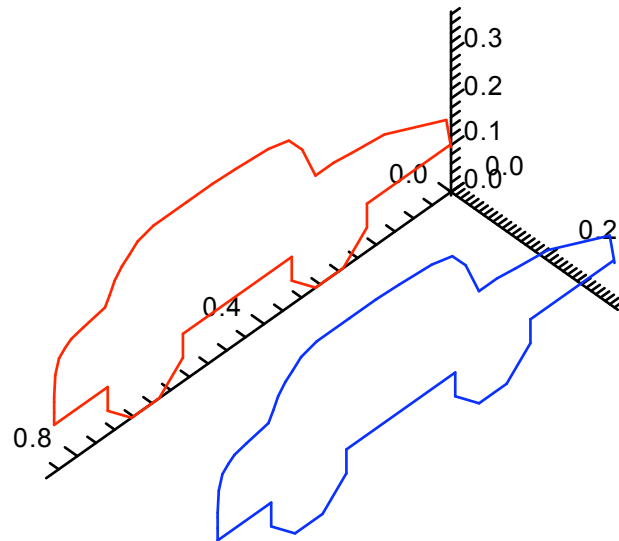
```
> listcarright := map(u-> <u[1], 0, u[2]>, listcar):
   listcarleft := map(u-> <u[1], 0.35, u[2]>, listcar):
```

Let's plot the two sides together, making the right side red and the left side blue. To plot the two sides with specified colors we use the display command. Note that the individual parts of the graph end with colons. The first plot shows two points in R^3 connected by a line segment (rotate the figure).

```
> pointplot3d(<1,2,3>, <2,3,4>, connect=true, axes=boxed, color=
   red);
```

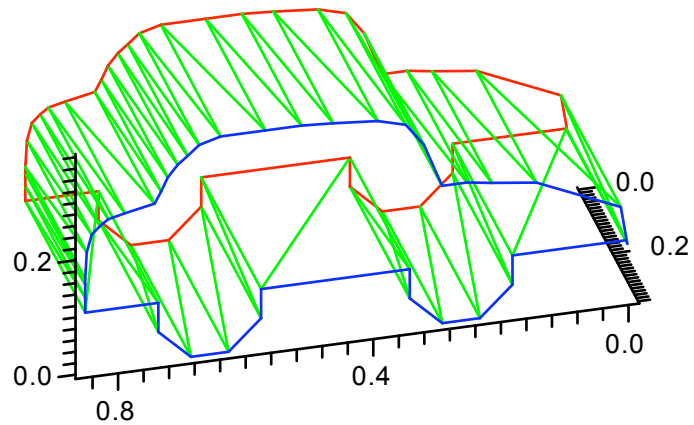


```
> rightside:= pointplot3d(listcarright, connect=true, color=red):  
leftside:= pointplot3d(listcarleft, color=blue, connect=true):  
display({rightside,leftside},axes=normal, scaling=constrained);
```



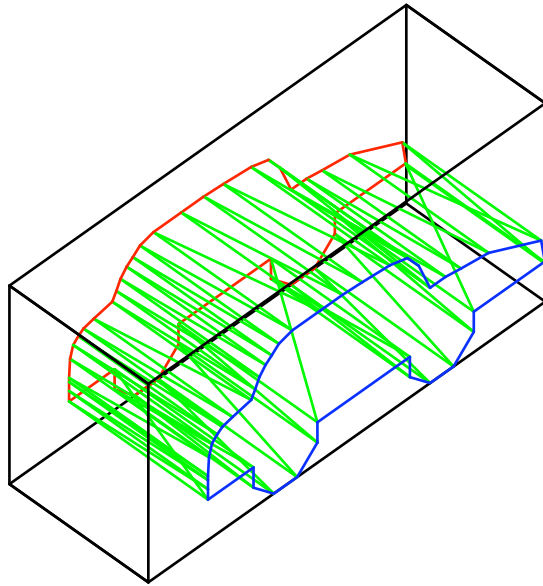
To make it look like a single object we draw lines connecting the two sides of the car. We do that in green.

```
> listcarcross := [seq(op([listcarright[k],listcarleft[k]]),k=1.
.41)]:
> crossing:= pointplot3d(listcarcross, color=green, connect=true)
:
display({leftside,rightside,crossing},axes=normal, scaling=
constrained);
```



Finally, we replace the axes with a garage. We will have transformations act on the car, while leaving the garage fixed so that it can act as a reference frame.

```
> garage:=[<0,0,0>, <1,0,0>, <1,.35,0>, <1,0,0>, <1,0,.5>, <1,.35,
.5>,
<1,0,.5>, <0,0,.5>, <0,.35,.5>, <0,0,.5>, <0,0,0>, <0,.35,
0>,
<1,.35,0>, <1,.35,.5>, <0,.35,.5>, <0,.35,0>]:
garage1:= pointplot3d(garage, color=black, connect=true):
display({leftside,rightside,crossing,garage1}, axes=None,
scaling=constrained);
```

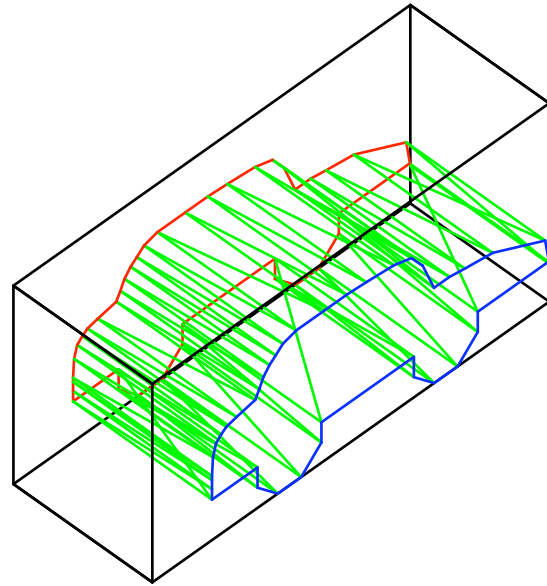


To reduce typing we define a function that turns a list of points into an outline of the desired color

```
> poly3d := (pointlist, shade) ->  
    pointplot3d(pointlist, style=line, color=shade, connect=  
    true):
```

This allows us plot the 4 lists together as follows.

```
> display(  
    {poly3d(listcarright, red), poly3d(listcarleft, blue),  
    poly3d(listcarcross, green), poly3d(garage, black)},  
    axes=None, scaling=constrained);
```



We use the 4 lists in this example for the remainder of the worksheet.

▼ Exercise

1) To check that you understand the syntax of the command, plot just the crossings of the car in orange together with the garage in green.

```
>
```

```
>
```

▼ Visualizing Linear Transformations in \mathbb{R}^3

In a previous worksheet we looked at 2 by 2 matrices acting on \mathbb{R}^2 . We are ready to extend this study to \mathbb{R}^3 . We borrow a procedure from that previous worksheet that allows us to multiply a matrix by a list of vectors.

```
> multmatbylist := proc(multmat, listofvecs)
    map(x->multmat.x,listofvecs);
end;
```

We start with a matrix that rotates objects by 90 degrees in the y-z plane (that is, around the x-axis). Such a transformation should send e_1 to e_1 , e_2 to e_3 , and e_3 to $-e_2$. We designate the matrix `rot1`.

```
> rot1:= <<1,0,0> | <0,0,1> | <0,-1,0>>;
testvec := <xcoord, ycoord, zcoord>;
rot1.testvec;
```

$$rot1 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

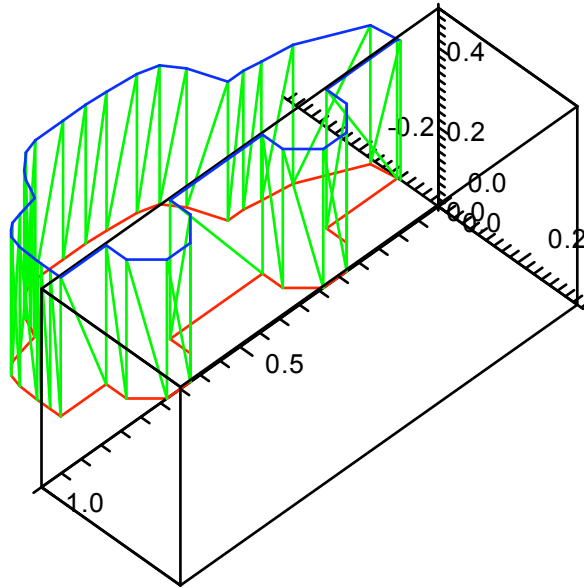
$$testvec := \begin{bmatrix} xcoord \\ ycoord \\ zcoord \end{bmatrix}$$

$$\begin{bmatrix} xcoord \\ -zcoord \\ ycoord \end{bmatrix}$$

(3.1)

Now we want to multiply the three carlists by rot1 and graph the result. Note that we leave the garage alone.

```
> rlist:=multmatbylist(rot1,listcarright):
l1list:=multmatbylist(rot1,listcarleft):
clist:=multmatbylist(rot1,listcarcross):
display(
  {poly3d(rlist,red), poly3d(l1list, blue),
   poly3d(clist, green), poly3d(garage, black)},
  axes=none, scaling=constrained);
```



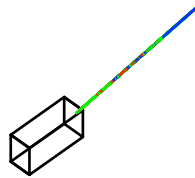
>

We see that the transformation turns the car on its side.

As a second example, shear the car in the z direction by the value of .3*x. (This jacks up the back end of the car.)

```
> shear1:= <<1, 2, 3> | <2, 4, 6> |<3, 6, 9>>;
rlist2:=multmatbylist(shear1,listcarright):
l1list2:=multmatbylist(shear1,listcarleft):
clist2:=multmatbylist(shear1,listcarcross):
display(
  {poly3d(rlist2,red), poly3d(l1list2, blue),
   poly3d(clist2, green), poly3d(garage, black)},
  axes=none, scaling=constrained);
```

$$shear1 := \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$$



Exercises:

2) Use matrix multiplication to produce a single matrix that composes the rotation in the y-z plane followed by the shearing in the x-z plane (think carefully about the order of multiplication) . Compare the result of multiplying this single matrix on the car with the result of first multiplying the car by the rotation matrix and then the result of the rotation by the shearing matrix.

>

3) Produce a matrix to rotate the figure counterclockwise 30 degrees about the x-axis and another matrix to rotate the image by 20 degrees counterclockwise about the z axis. Apply these matrices in succession to the image of the car and plot the result.

>

4) Obtain a single matrix that has the same effect as the composite of the two matrices in Exercise 3. Plot the result.

>

5) Write a hypothesis about the result of multiplying the image of the car by a rank 2 matrix. Create a random rank 2 matrix of size 3 by 3 and verify your hypothesis. (Your matrix should be of rank 2 with no zeros. Pay particular attention to what happens to the images of the sides of

the car.)

>

6) Write a hypothesis about the result of multiplying the image of the car by a matrix with all zeros in the third column. Create a random 3 by 3 matrix of this type and verify your hypothesis. (Your matrix should be of rank 2 with zeros only in the third column. Pay particular attention to what happens to the images of the sides of the car. Your hypothesis should mention vectors that are in the null space.)

>

7) Write a hypothesis about the result of multiplying the image of the car by a rank 1 matrix. Create a random rank 1 matrix of size 3 by 3 and verify your hypothesis.

>

8) Write a hypothesis about the result of multiplying the image of the car by a matrix with all zeros in the first and third rows. Create a random 3 by 3 matrix of this type and verify your hypothesis. (Your hypothesis should mention interesting vectors in the image.)

>

>

>