Symmetries of an Equilateral Triangle

Adapted from Visualizing Linear Transformations in R^2 *(May and Blyth)*
by Dorothy Zeiser, Gwynedd Mercy College

```
> 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

1. *Introduction*
2. *Review matrix representation for rotations around the origin.*
3. *Review matrix representation for reflections in a line.*
4. *Represent the edges of an equilaterial triangle in matrix form.*
5. *Perform and display the results of rotations.*
6. *Perform and display the results of reflections.*
7. *Perform and display composition of pairs of transformations.*
8. *Comparison with rotations and reflections done via plottools.*

Introduction

This worksheet is intended for use in the beginning of an Abstract Algebra class to help students to visualize the results of rotations and reflections that return the transformed triangle to the same area occupied by the original triangle.

Rotations around the Origin

An important example discussed in the text is rotation around the origin by angle α radians. We see that this rotation takes [1, 0] to [cos(α), sin(α)] and takes [0, 1] to [-sin (α), cos(α)]. This allows us to produce a matrix that can be used for the rotation (these two images are placed into the matrix as its columns). Note alpha must be expressed in radians.

```
> rotmat := alpha -> <<cos(alpha), sin(alpha)> | <-sin(alpha), cos
  (alpha)>>;
```

$$rotmat := \alpha \rightarrow \langle\langle \cos(\alpha), \sin(\alpha)\rangle | \langle -\sin(\alpha), \cos(\alpha)\rangle\rangle \quad (3.1)$$

In an equilateral triangle there are three rotations that leave the triangle occupying the same area in the plane: 0 degrees, 120 degrees, and 240 degrees.
Define these as

```
> Rot0 := rotmat(0);
  Rot120 := rotmat(2*Pi/3);
  Rot240 := rotmat(4*Pi/3);
```

$$Rot0 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Rot120 := \begin{bmatrix} \frac{-1}{2} & -\frac{1}{2}\sqrt{3} \\ \frac{1}{2}\sqrt{3} & \frac{-1}{2} \end{bmatrix}$$

$$Rot240 := \begin{bmatrix} \frac{-1}{2} & \frac{1}{2}\sqrt{3} \\ -\frac{1}{2}\sqrt{3} & \frac{-1}{2} \end{bmatrix} \qquad (3.2)$$

Reflections in a Line

In an equilateral triangle there are three reflections that leave the triangle occupying the same area in the plane. They are the reflections in the lines containing an altitude of the equilateral triangle. We can position our triangle so that one of the altitudes lies along the y-axis.

A reflection in the y-axis takes [1, 0] to [-1,0] and takes [0, 1] to [0,1]. This allows us to produce a matrix that can be used for the reflection (these two images are placed into the matrix as its columns).

```
> VR := Matrix(<< -1 | 0 >,
          < 0 | 1 >>);
```

$$VR := \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (4.1)$$

(The other two matrices have not been developed yet. It does not seem worth it when students can see the results thus far with matrices. It is much easier with plottools.)

```
>
```

Define and Display an Equilateral Triangle

1. Define each edge of the equilateral triangle.

Matrix representation of each edge needed for matrix multiplication.

```
> ET1list := [<1,-sqrt(3)/3>,<0,2*sqrt(3)/3>];
  ET2list := [<0,2*sqrt(3)/3>,<-1,-sqrt(3)/3>];
  ET3list := [<-1,-sqrt(3)/3>,<1,-sqrt(3)/3>];
```

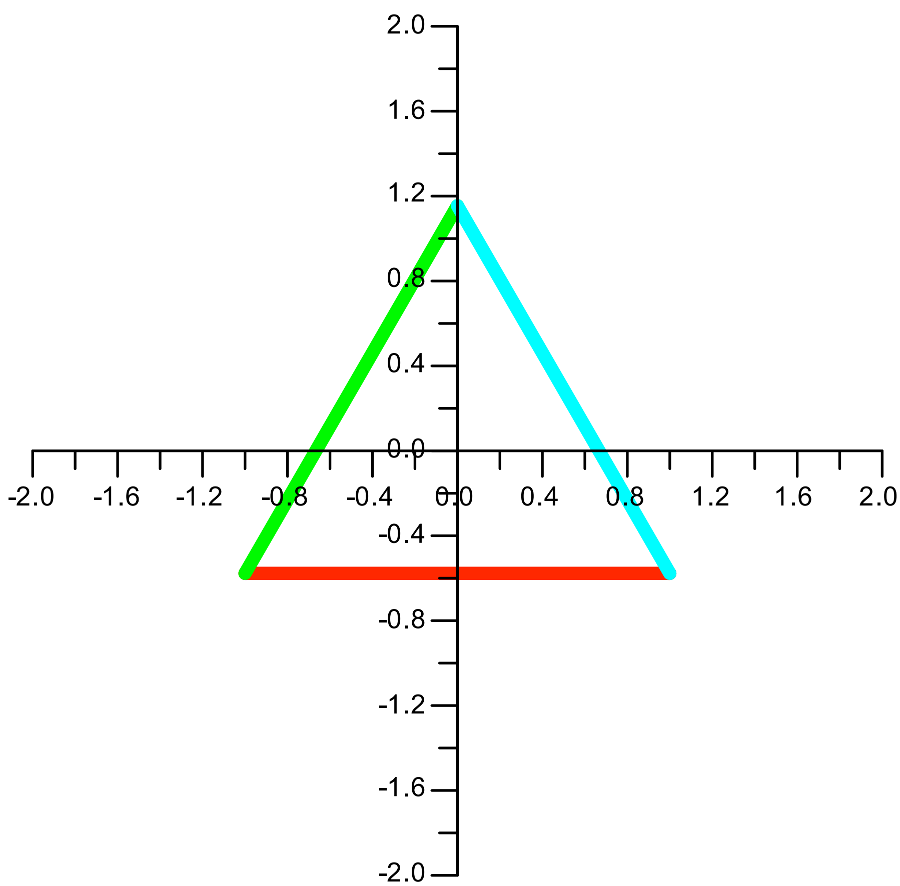
$$ET1list := \left[\begin{bmatrix} 1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{2}{3}\sqrt{3} \end{bmatrix} \right]$$

$$ET2list := \left[\left[\begin{array}{c} 0 \\ \frac{2}{3}\sqrt{3} \end{array}\right], \left[\begin{array}{c} -1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right]\right]$$

$$ET3list := \left[\left[\begin{array}{c} -1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right], \left[\begin{array}{c} 1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right]\right] \qquad (5.1.1)$$

2. Display the triangle in initial position.

```
> ET1 := pointplot(ET1list, view=[-2..2, -2..2],connect=true,
  color=cyan,thickness=5,axes=normal):
  ET2 := pointplot(ET2list, view=[-2..2, -2..2],connect=true,
  color=green,thickness=5,axes=normal):
  ET3 := pointplot(ET3list, view=[-2..2, -2..2],connect=true,
  color=red,thickness=5,axes=normal):
> display(ET1,ET2,ET3);
```



```
>
```

Linear Transformations in R^2

Mappings of vector spaces that satisfy the linearity properties can be represented as multiplication by a matrix. We follow the text's convention, treating vectors as column vectors so that the matrix is always on the left and the vector is on the right.

```
> A := <<1,-2> | <0,1>>;
  b := <1,3>;
  "A b " =A.b;
```

$$A := \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}$$

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

$$\text{"A b "} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (6.1)$$

To see how multiplication by a matrix changes a figure, we apply the matrix to a list of vectors.

```
> multmatbylist :=(multmat, listofvecs)->
    map((x,y)-> y.x,listofvecs,multmat):
```

Counterclockwise Rotations around the Origin

1. Rotation of 120 degrees

Apply a rotation of 120 to each of the three edges of the triangle.

```
> R120_ET1_list := multmatbylist(Rot120,ET1list);
  R120_ET2_list := multmatbylist(Rot120,ET2list);
  R120_ET3_list := multmatbylist(Rot120,ET3list);
```

$$R120\_ET1\_list := \left[\begin{bmatrix} 0 \\ \frac{2}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} -1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix} \right]$$

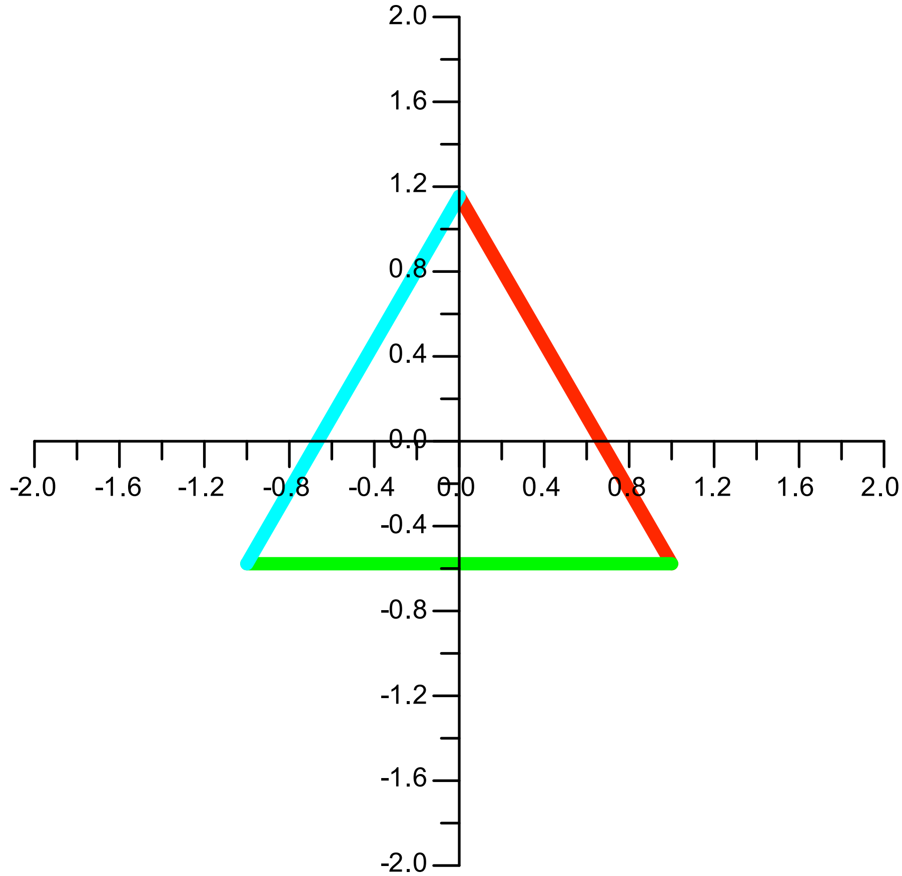
$$R120\_ET2\_list := \left[\begin{bmatrix} -1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} 1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix} \right]$$

$$R120\_ET3\_list := \left[\begin{bmatrix} 1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{2}{3}\sqrt{3} \end{bmatrix} \right] \quad (6.1.1.1)$$

Prepare plots and display triangle after 120 degree rotation.

```
> R120_ET1_edge := pointplot(R120_ET1_list, view=[-2..2, -2.
  .2],connect=true,color=cyan,thickness=5,axes=normal):
  R120_ET2_edge := pointplot(R120_ET2_list, view=[-2..2, -2.
  .2],connect=true,color=green,thickness=5,axes=normal):
  R120_ET3_edge := pointplot(R120_ET3_list, view=[-2..2, -2.
```

```
.2],connect=true,color=red,thickness=5,axes=normal):
> display(R120_ET1_edge,R120_ET2_edge,R120_ET3_edge);
```



Exercise: Copy, paste, and modify the above three sections of maple input to perform a rotation of 240 degrees and display the resulting figure.

Apply a rotation of 240 to each of the three edges of the triangle.

```
>
```

Prepare plots and display triangle after 240 degree rotation.

```
>
>
```

Reflections in the Lines containing theAltitudes

Reflection in the y-axis

Apply a reflection in the y-axis to each of the three edges of the triangle.

```
> VR_ET1_list := multmatbylist(VR,ET1list);
  VR_ET2_list := multmatbylist(VR,ET2list);
  VR_ET3_list := multmatbylist(VR,ET3list);
```

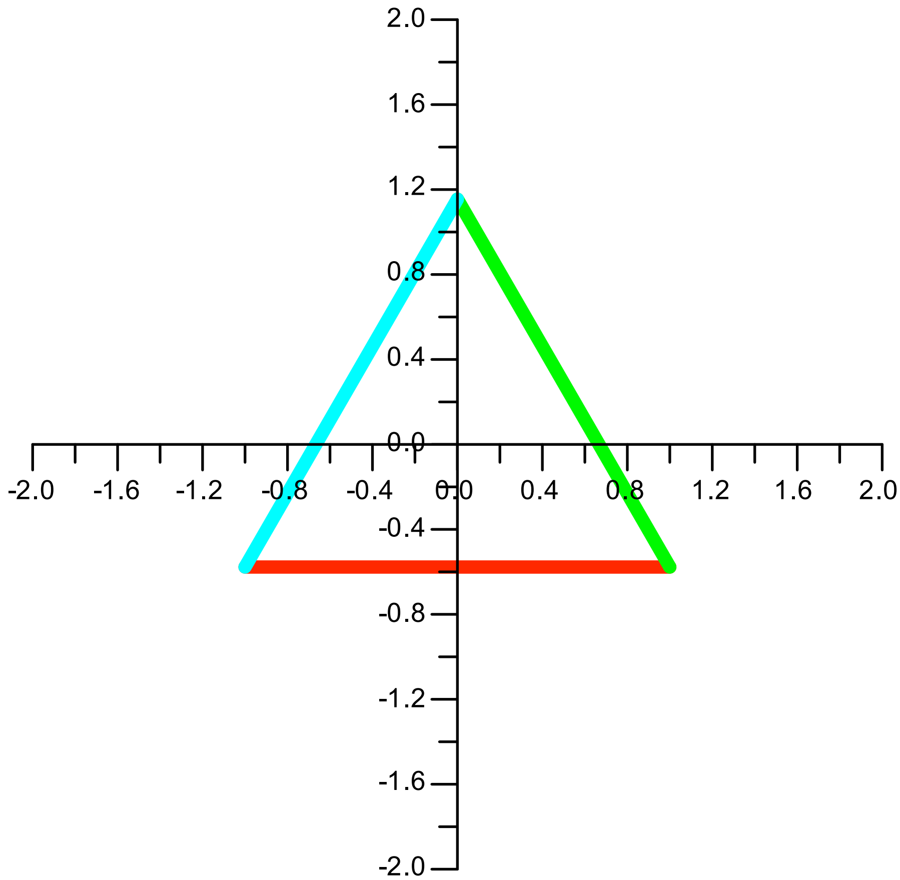
$$VR\_ET1\_list := \left[\left[\begin{array}{c} -1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right], \left[\begin{array}{c} 0 \\ \frac{2}{3}\sqrt{3} \end{array}\right]\right]$$

$$VR\_ET2\_list := \left[\left[\begin{array}{c} 0 \\ \frac{2}{3}\sqrt{3} \end{array}\right], \left[\begin{array}{c} 1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right]\right]$$

$$VR\_ET3\_list := \left[\left[\begin{array}{c} 1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right], \left[\begin{array}{c} -1 \\ -\frac{1}{3}\sqrt{3} \end{array}\right]\right] \quad (6.2.1)$$

Prepare plots and display triangle after reflection in y-axis.

```
> VR_ET1_edge := pointplot(VR_ET1_list, view=[-2..2, -2..2],
  connect=true,color=cyan,thickness=5,axes=normal):
  VR_ET2_edge := pointplot(VR_ET2_list, view=[-2..2, -2..2],
  connect=true,color=green,thickness=5,axes=normal):
  VR_ET3_edge := pointplot(VR_ET3_list, view=[-2..2, -2..2],
  connect=true,color=red,thickness=5,axes=normal):
> display(VR_ET1_edge,VR_ET2_edge,VR_ET3_edge);
```



>

Composition of Transformations

Example: Apply a rotation of 120 degrees followed a reflection in the y-axis.

Apply a reflection **in** *the y−axis* **to** *the result of a rotation of* 120 *for each of the three edges of the triangle.*

```
> VR_R120_ET1_list := multmatbylist(VR,R120_ET1_list);
  VR_R120_ET2_list := multmatbylist(VR,R120_ET2_list);
  VR_R120_ET3_list := multmatbylist(VR,R120_ET3_list);
```

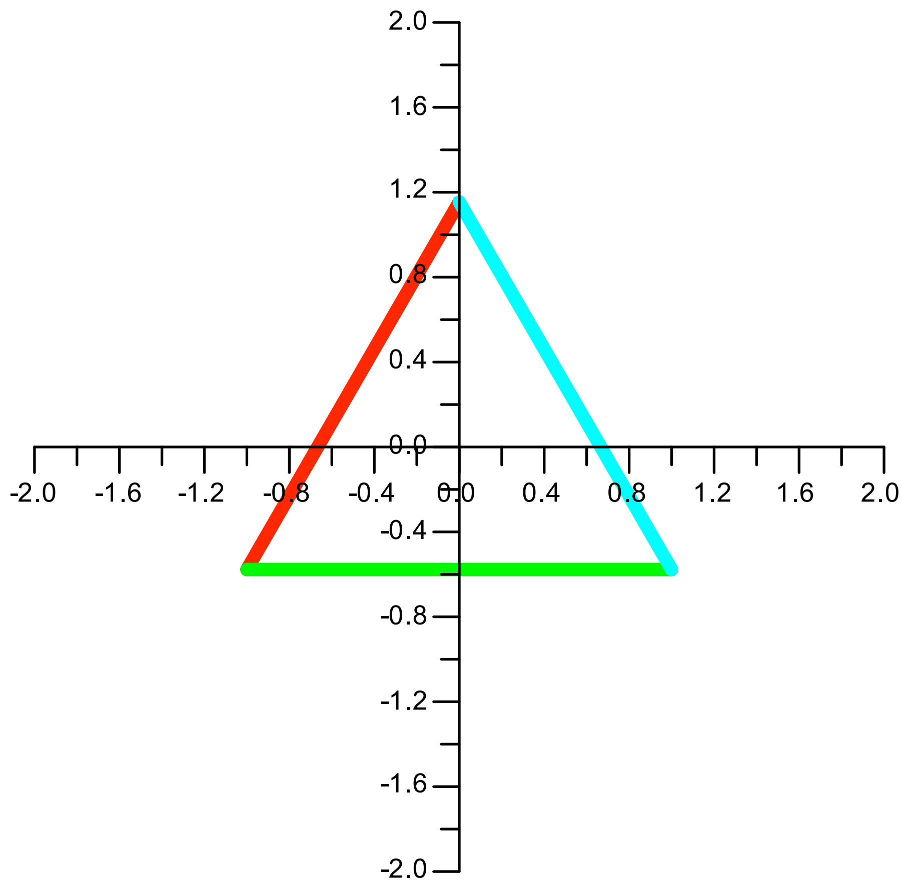
$$VR\_R120\_ET1\_list := \left[\begin{bmatrix} 0 \\ \frac{2}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} 1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}\right]$$

$$VR\_R120\_ET2\_list := \left[\begin{bmatrix} 1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} -1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix} \right]$$

$$VR\_R120\_ET3\_list := \left[\begin{bmatrix} -1 \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{2}{3}\sqrt{3} \end{bmatrix} \right] \qquad (6.3.1.1)$$

Prepare plots and display triangle after rotation of 120 followed by reflection in y-axis.

```
> VR_R120_ET1_edge := pointplot(VR_R120_ET1_list, view=[-2..2, -2..2],connect=true,color=cyan,thickness=5,axes=normal):
  VR_R120_ET2_edge := pointplot(VR_R120_ET2_list, view=[-2..2, -2..2],connect=true,color=green,thickness=5,axes=normal):
  VR_R120_ET3_edge := pointplot(VR_R120_ET3_list, view=[-2..2, -2..2],connect=true,color=red,thickness=5,axes=normal):
> display(VR_R120_ET1_edge,VR_R120_ET2_edge,VR_R120_ET3_edge);
```




```
>
```

Comparison with results via plottools

Done in a separate worksheet.

>