```
> restart;
> interface(warnlevel = 0) :   #   Maple 12
> with(plots) :
> with(LinearAlgebra) :
```

Vector rotation in 3D

Here we define the three rotation operators; the rotation matrices Rx(θ), Ry(θ), and Rz(θ)

```
> Rz := θ → Matrix([[cos(θ), − sin(θ), 0], [sin(θ), cos(θ), 0], [0, 0, 1]]) :
  Rx := θ → Matrix([[1, 0, 0], [0,cos(θ), − sin(θ)], [0,sin(θ), cos(θ)],]) :
  Ry := θ → Matrix([[cos(θ), 0, sin(θ)], [0, 1, 0], [− sin(θ), 0, cos(θ)]]) :
  'Rz(θ)'= Rz(θ); 'Rx(θ)'= Rx(θ); 'Ry(θ)'= Ry(θ);
```

$$Rz(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Rx(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$Ry(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad \textbf{(1)}$$

The following procedure shows the rotation cone as a vector rotates in 3D space about the x, y, and z axis of a Cartesian coordinate system

cones(s,V)

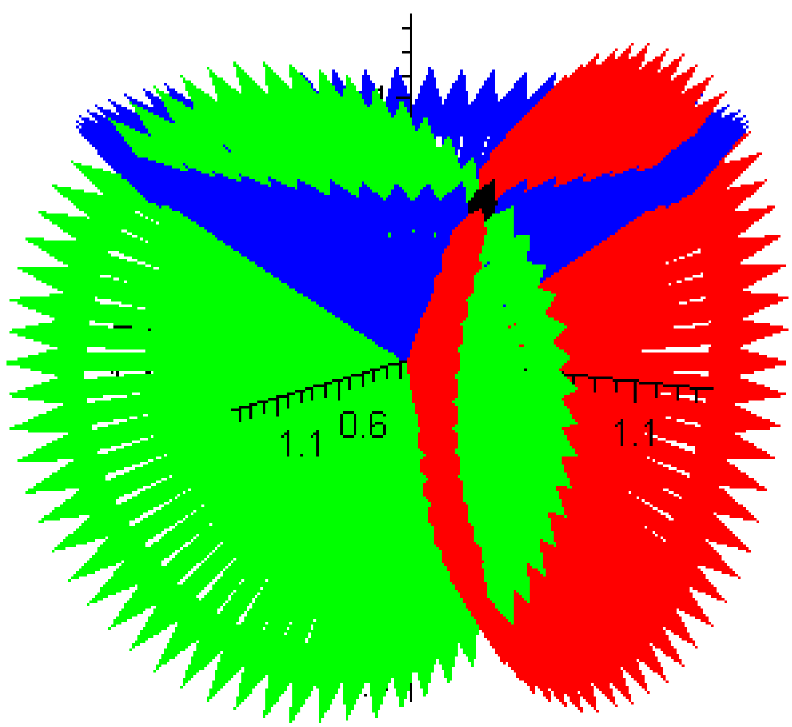
where s is the rotation step; increment

V is the vector to be rotated

```
> cone := proc(s, V)
        local  i, r, X, Y, Z, Vx, Vy, Vz, Vxi, Vyi, Vzi, P, θ;
       X := [ ];  Y := [ ];  Z := [ ];
       r := evalf(s);
       for i from 0 by r to 2 do
           θ := π·i;                          #``full rotation − 360 degrees, 2π
           Vzi := simplify(Multiply(Rz(θ), V)) :
           Vxi := simplify(Multiply(Rx(θ), V)) :  # incremental vectors
           Vyi := simplify(Multiply(Ry(θ), V)) :
           Z := [op(Z),Vzi];
           X := [op(X), Vxi]; # list of vectors
           Y := [op(Y), Vyi];
       end do;
       Vz := arrow(Z, color = blue);
       Vx := arrow(X, color = green); # coloring the rotating vectors
       Vy := arrow(Y, color = red);
        P := arrow(V, color = black);   # the original vector
       display([Vx, Vy, Vz, P],  axes = normal, scaling = constrained, orientation = [31, 80]);
   end proc :
```

The vector to be rotated about the axes is V = i + j + k ; a 3D vector. Steps of rotation is 1/32.

> $cone\left(\frac{1}{32}, < 1, 1, 1 >\right);$



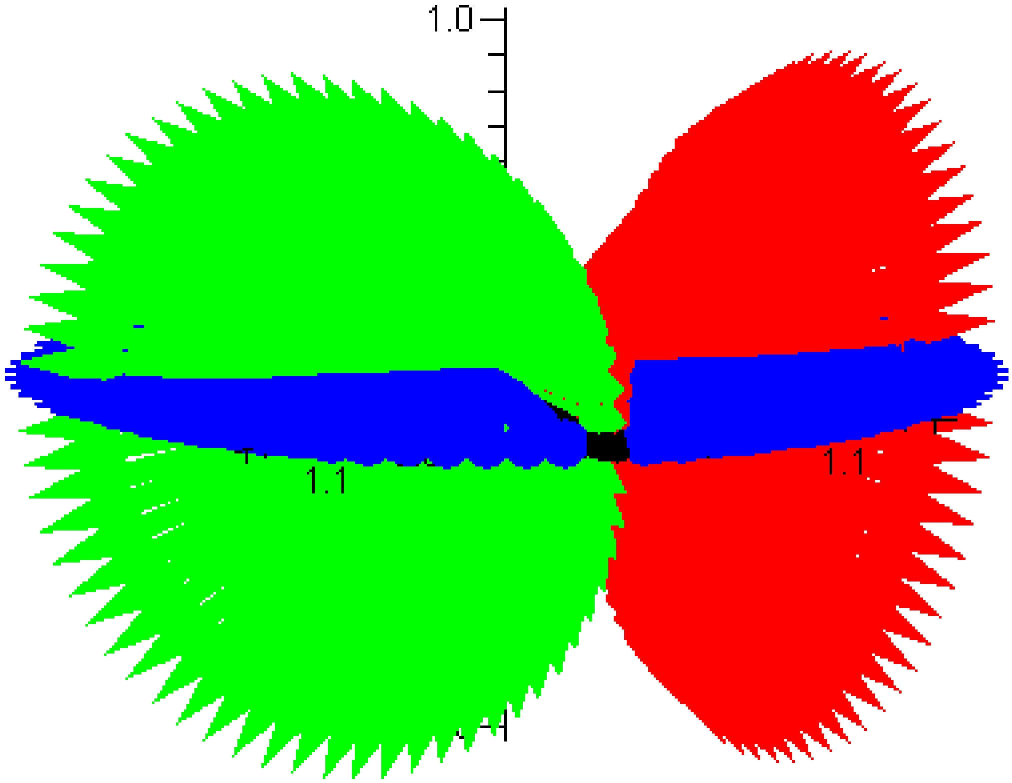
The next vector to be rotated about the axes is

V = i + j ; a vector in lying in the x-y plane.

Hence rotating this vector about the z axis traces out a circle.

Again, the step of rotation is 1/32

> $cone\left(\frac{1}{32}, < 1, 1, 0 >\right);$



The following procedure illustrates a sequence of rotations of a vector in 3D space. The first rotation is about the z axis, the second rotation is about the x axis, and the final rotation is about the y axis.

seqr(a1,a2,a3,V,s)

where a1 is the angle of rotation about the z-axis
a2 is the angle of rotation about the x-axis
a3 is the angle of rotation about the y-axis
V is the vector to be rotated
s is the rotation step

```
> seqr :=proc(a1, a2, a3, V, s)
      local i, r, last, X, Y, Z, Vx, Vy, Vz, Vxi, Vyi, Vzi, P, P1, P2, P3, θ;
     X := [ ]; Y := [ ]; Z := [ ];
     P := arrow(V, color = black); # initial vector
     last := a1/180;
     for i from 0 by s to last do
          θ := π· i;
          Vzi := simplify(Multiply(Rz(θ), V)) : # rotation about z axis
            Z := [op(Z),Vzi];
     end do;
     P1 := arrow(Vzi, color = magenta); # rotated vector P about z
     last := a2/180;
     for i from 0 by s to last do
          θ := π· i;
            Vxi := simplify(Multiply(Rx(θ), Vzi)) : # rotation about x axis
            X := [op(X), Vxi];
      end do;
      P2 := arrow(Vxi, color = yellow); # rotated vector P1 about x
      last := a3/180;
      for i from 0 by s to last do
            θ := π· i;
              Vyi := simplify(Multiply(Ry(θ), Vxi)) : # rotation about y axis
             Y := [op(Y), Vyi];
      end do;
      P3 := arrow(Vyi, color = brown); # rotated vector P2 about y
      Vz := arrow(Z, color = blue);            # coloring the rotations
     Vx := arrow(X, color = green);
     Vy := arrow(Y, color = red);
      print(Vi = V, Vf = Vyi);
      return [Vx, Vy, Vz, P, P1, P2, P3] ;
 end proc:
```

The vector to be rotated is

V = i + j ; a vector lying in the x-y plane

first rotation is 180 degrees about the z-axis

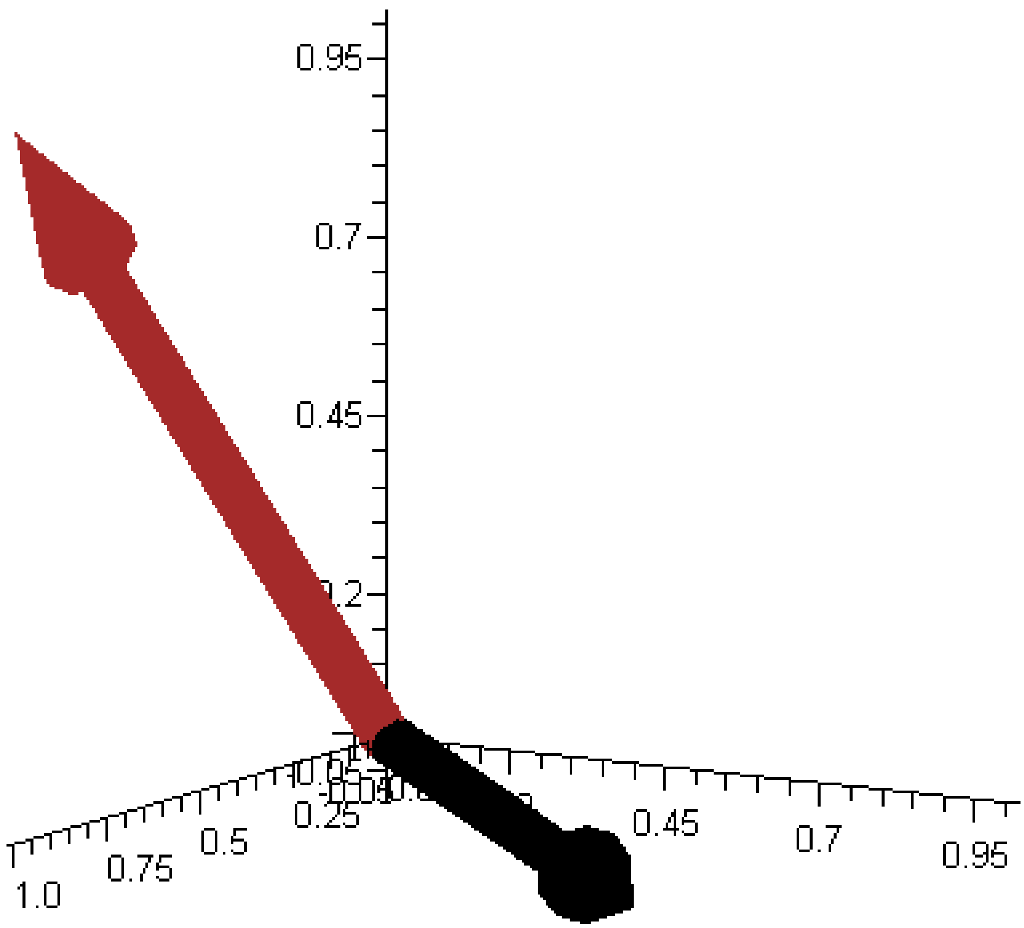
second rotation is 90 degrees about the x-axis

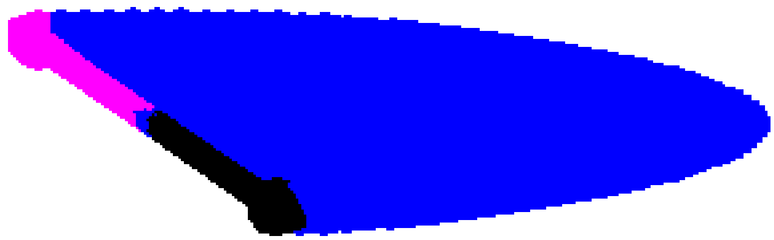
final rotation is 180 degrees about the y-axis

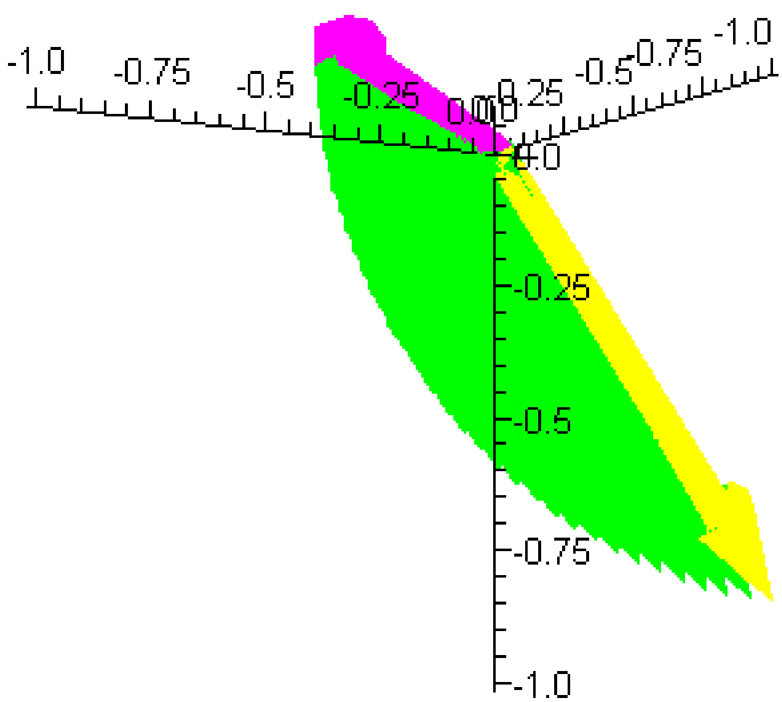
> $L := seqr\left(180, 90, 180, \langle 1, 1, 0\rangle, \frac{1}{64}\right):$

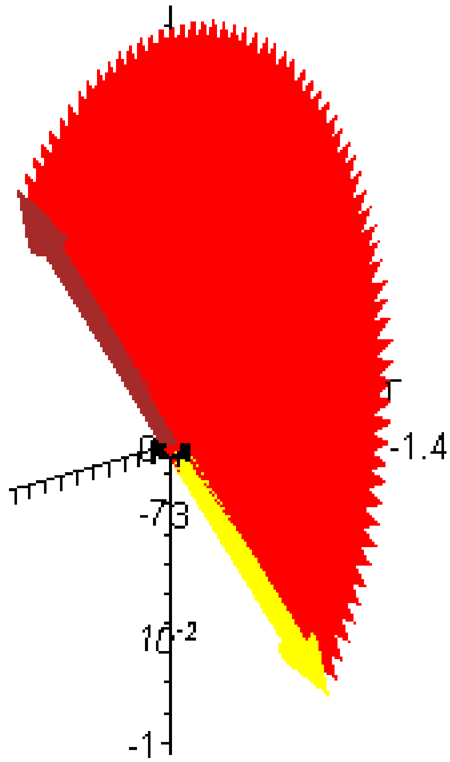
display([L[4], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[3], L[4], L[5]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[1], L[5], L[6]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[2], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[1], *L*[2], *L*[3], *L*[4], L[5], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);

$$Vi = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, Vf = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

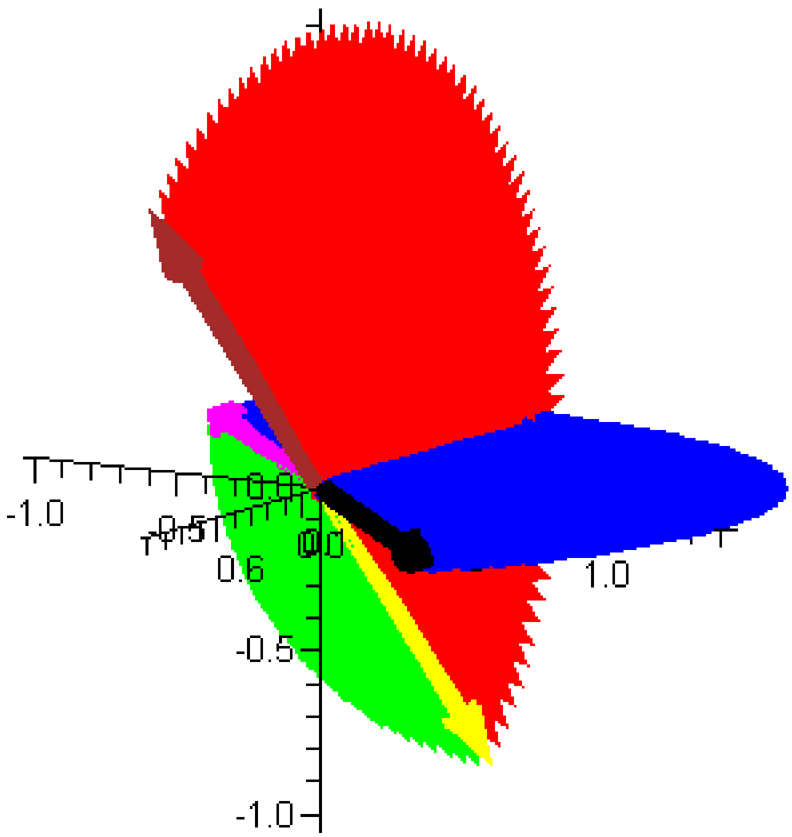





-1.0
-0.75
-0.5
-0.25
-0.5
-0.75
-1.0
-0.25
-0.5
-0.75
-1.0



-1.4
-7.3
10^-2
-1



-1.0
-0.5
0.0
1.0
0.6
-0.5
-1.0

The vector to be rotated is

V = i + j + k; a vector in 3D

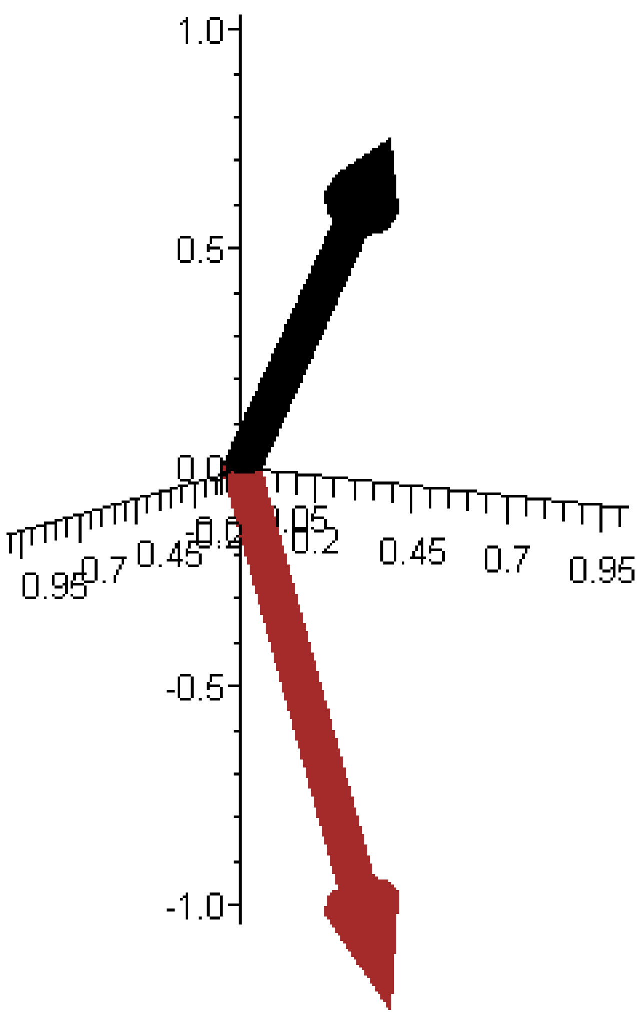
first rotation is 180 degrees about the z-axis

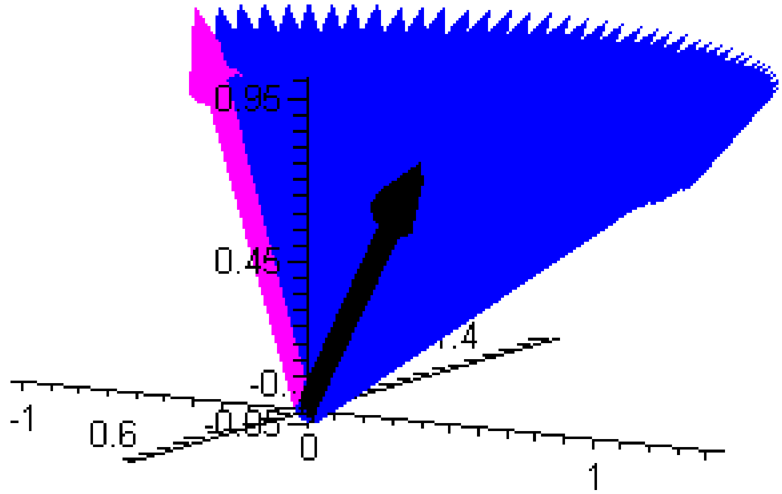
second rotation is 270 degrees about the x-axis

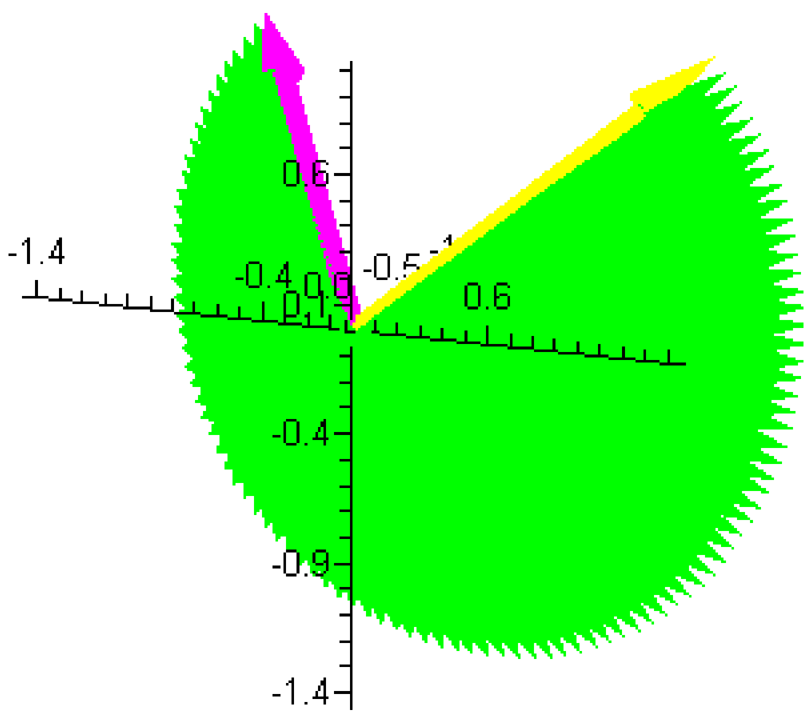
final rotation is 180 degrees about the y-axis

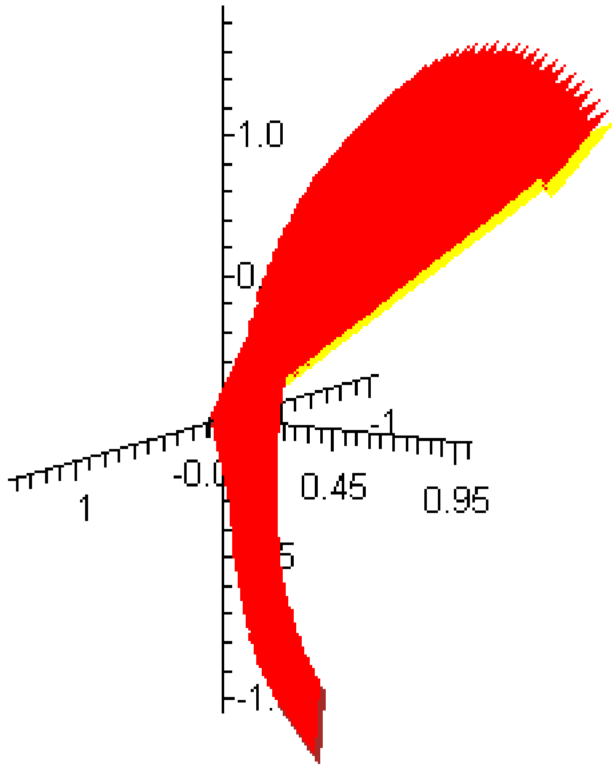
```
> L := seqr(180, 270, 180, ⟨1, 1, 1⟩, 1/64) :
display([L[4], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[3], L[4], L[5]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[1], L[5], L[6]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[2], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[1], L[2], L[3], L[4], L[5], L[6], L[7]], axes = normal, scaling = constrained, orientation
    = [31, 80]);
```

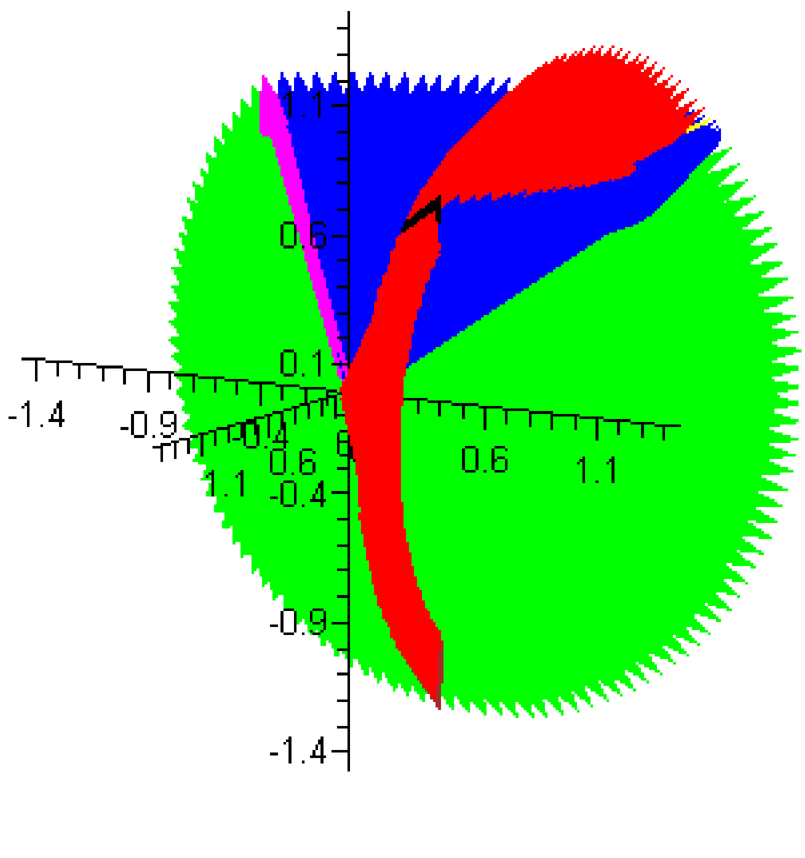
$$Vi = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, Vf = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$




0.95
0.45
-1
0.6
0
1


0.6
-1.4
0.6
-0.4
-0.9
-1.4


1.0
1
0.45
0.95
-1


1.1
0.6
0.1
-1.4
-0.9
-0.4
0.6
1.1
-0.4
-0.9
-1.4

The same vector

V = i + j + k; a vector in 3D

first rotation is 180 degrees about the z-axis

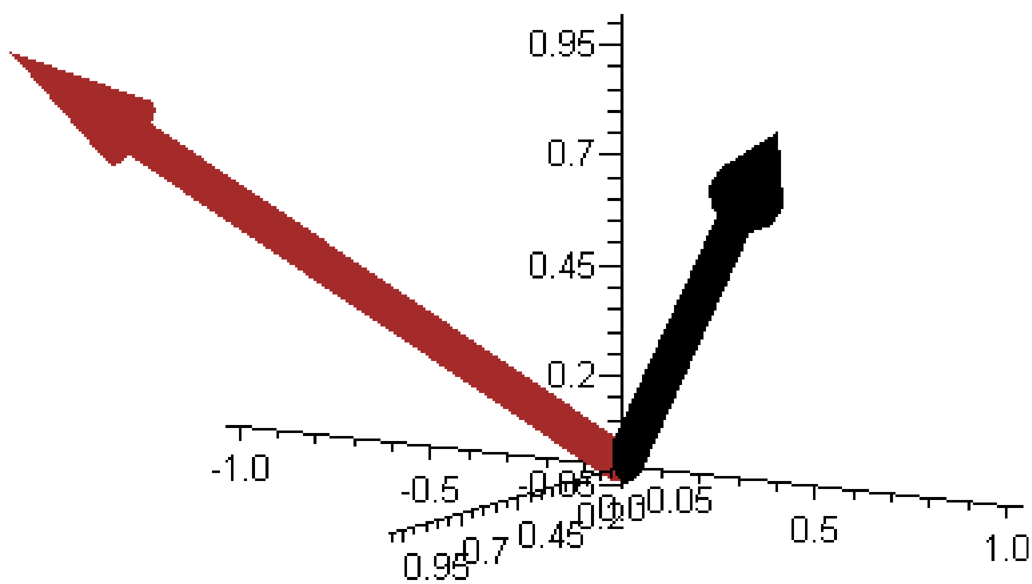
second rotation is 90 degrees about the x-axis

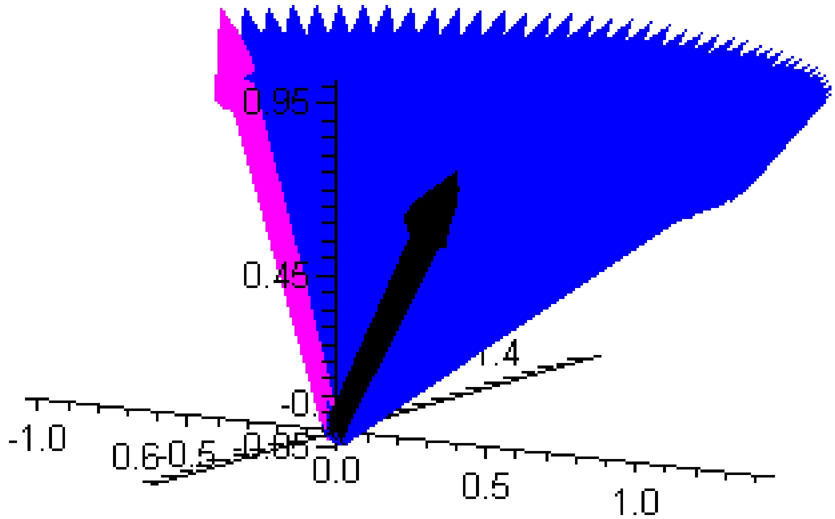
final rotation is 180 degrees about the y-axis

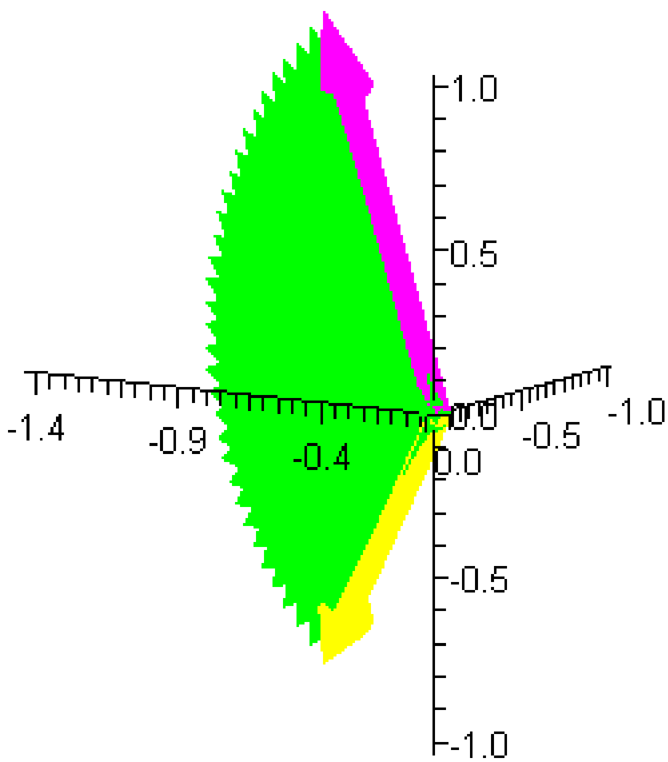
$> L := seqr\left(180, 90, 180, \langle 1, 1, 1\rangle, \frac{1}{64}\right):$

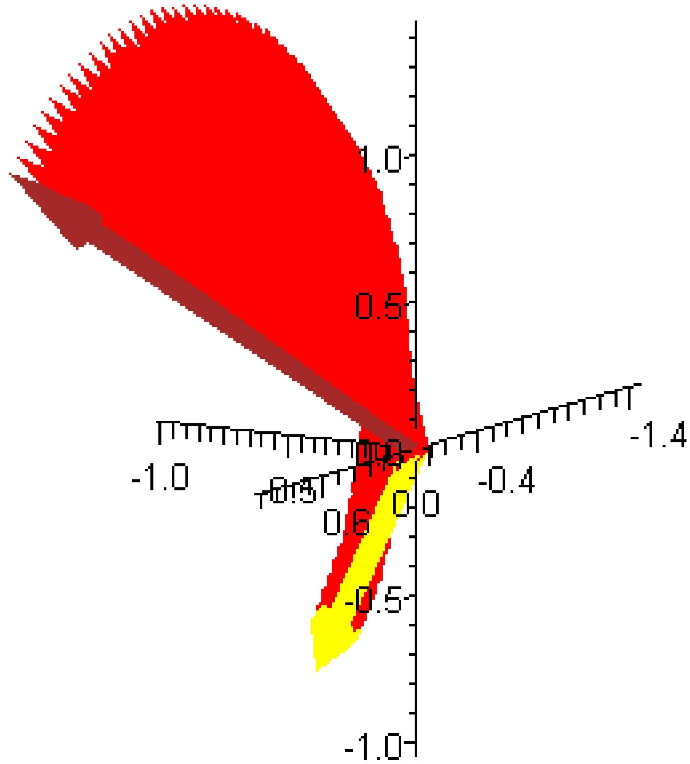
display([L[4], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[3], L[4], L[5]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[1], L[5], L[6]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[2], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);
display([L[1], *L*[2], *L*[3], *L*[4], L[5], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);

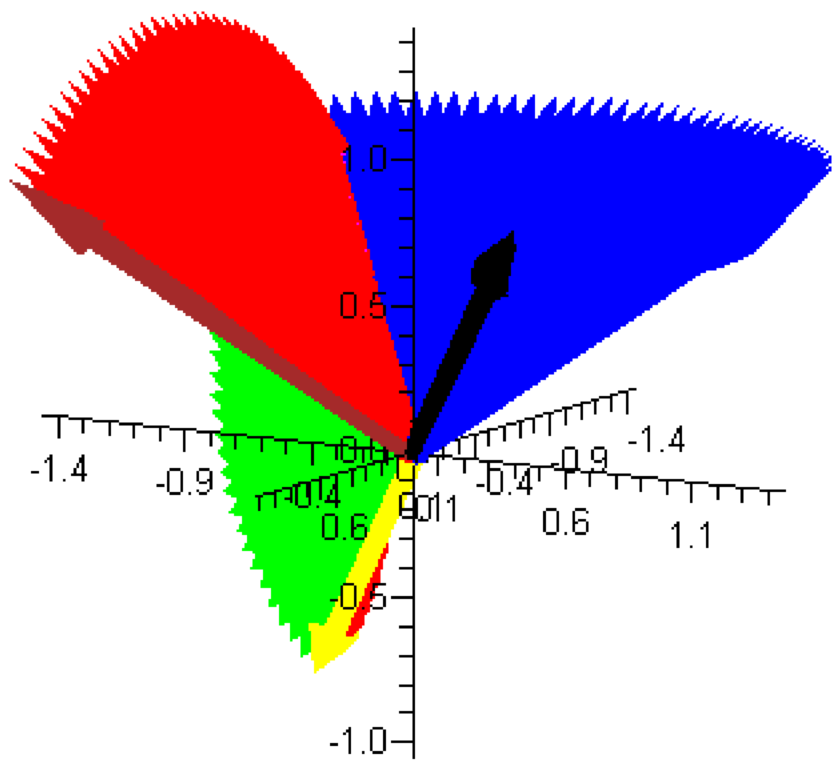
$$Vi = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, Vf = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$




0.95
0.45
1.4
-1.0
0.6
0.5
0.0
0.5
1.0


1.0
0.5
-1.4
-0.9
-0.4
0.0
-0.5
-1.0
0.0
-0.5
-1.0


1.0
0.5
-1.0
-0.5
0.0
-0.4
-1.4
0.6
-1.0
-0.5
-1.0


1.0
0.5
-1.4
-0.9
-0.4
-1.4
-0.9
-0.4
0.6
1.1
0.6
-0.5
-1.0

Same vector

V = i + j + k; a vector in 3D

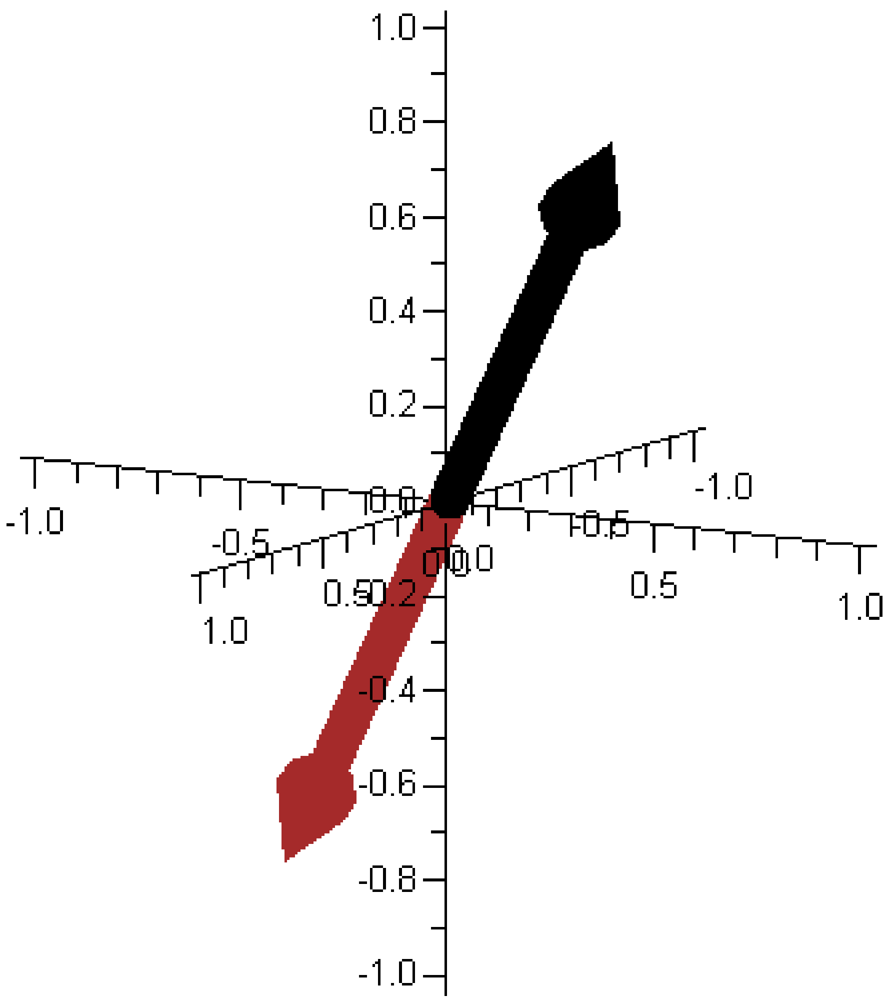
first rotation is 90 degrees about the z-axis

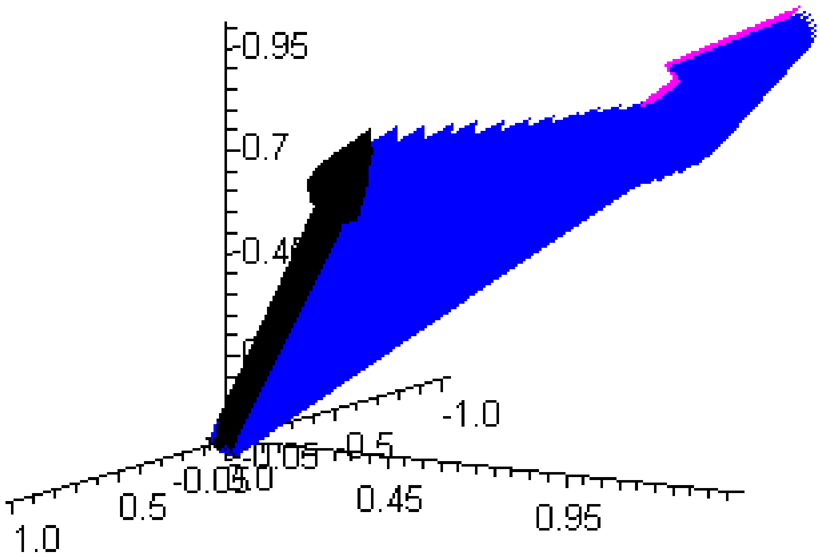
second rotation is 180 degrees about the x-axis

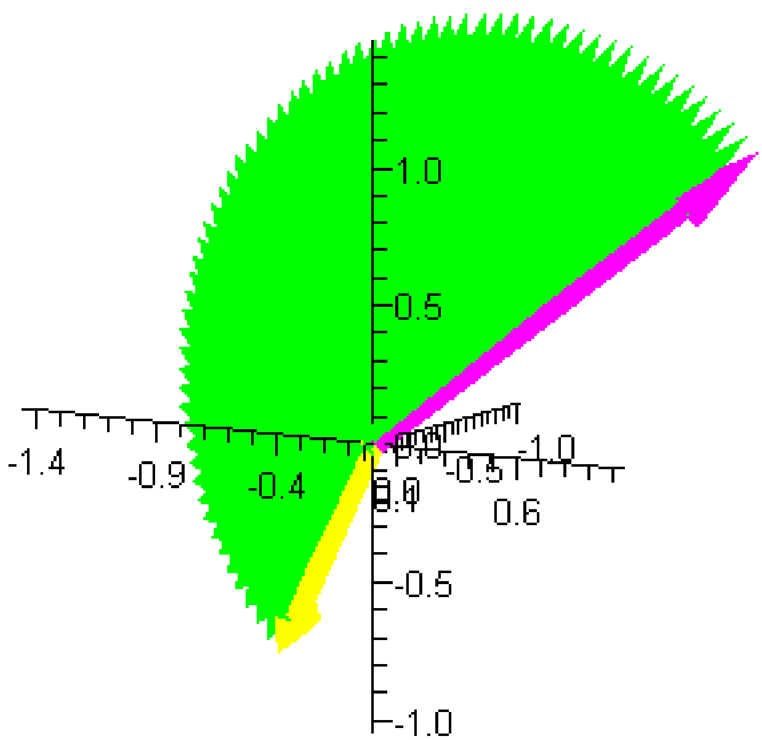
no rotation about the y-axis

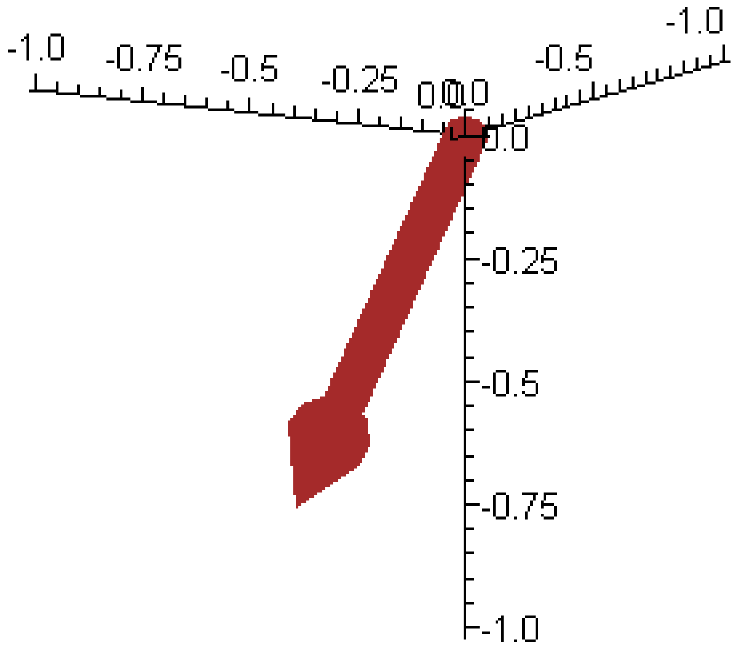
> $L := seqr\left(90, 180, 0, \langle 1, 1, 1\rangle, \frac{1}{64}\right):$
> $display([L[4], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);$
> $display([L[3], L[4], L[5]], axes = normal, scaling = constrained, orientation = [31, 80]);$
> $display([L[1], L[5], L[6]], axes = normal, scaling = constrained, orientation = [31, 80]);$
> $display([L[2], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);$
> $display([L[1], L[2], L[3], L[4], L[5], L[6], L[7]], axes = normal, scaling = constrained, orientation = [31, 80]);$

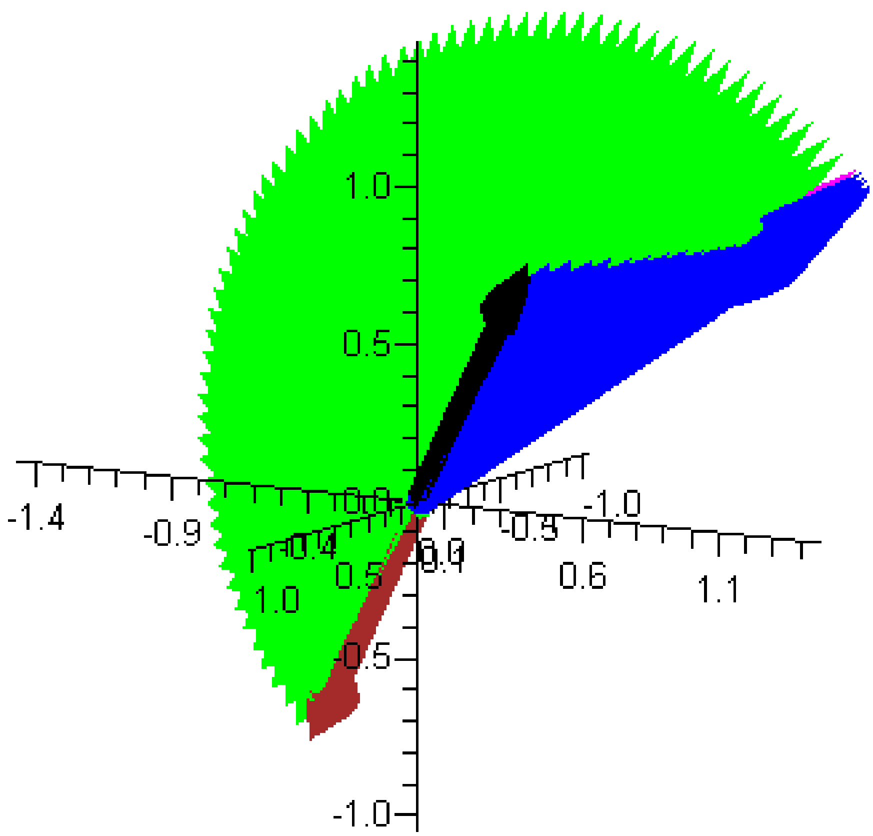
$$Vi = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, Vf = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$




0.95
0.7
-1.0
-0.5
-0.05
0.45
0.95
0.5
1.0


1.0
0.5
-1.4
-0.9
-0.4
-1.0
0.6
-0.5
-1.0





Notice that none of the rotations changed the length of the vectors.