

# HW1

Donald Butler

01/29/2022

## Create plot of Initials

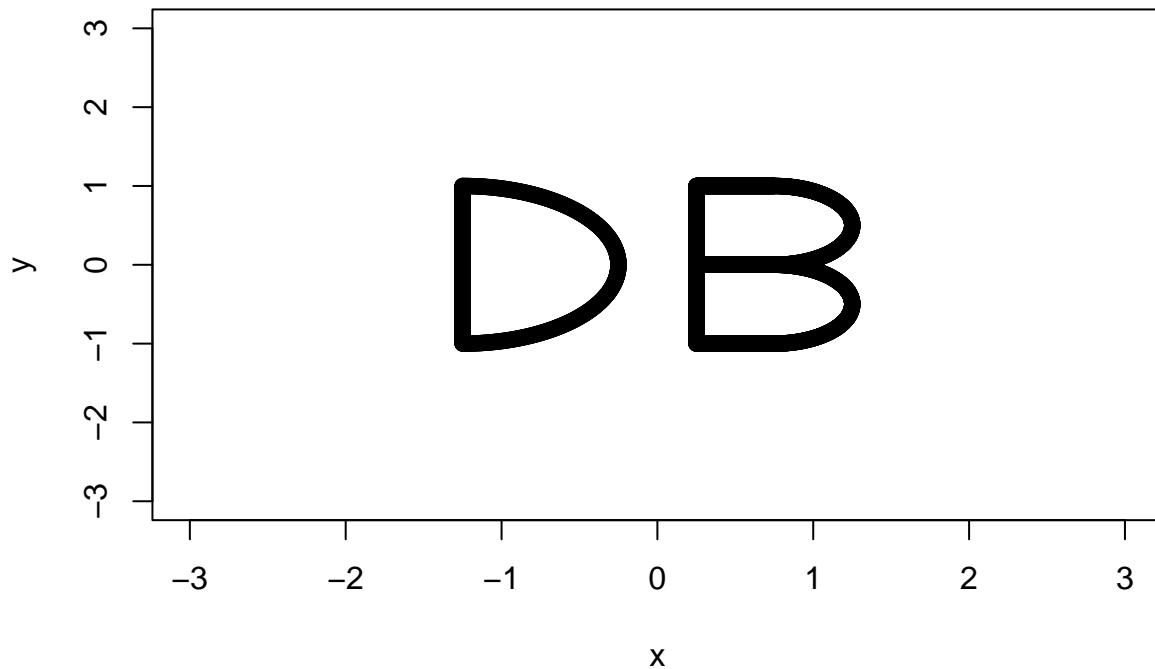
Create a plot of the initials in my first and last name.

```
x=c(rep(-1.25,500),
  cos(seq(-pi/2,pi/2,length.out=500))-1.25,
  rep(.25,500),
  seq(.25,.75,length.out=500),
  seq(.25,.75,length.out=500),
  seq(.25,.75,length.out=500),
  .5*cos(seq(-pi/2,pi/2,length.out=500))+.75,
  .5*cos(seq(-pi/2,pi/2,length.out=500))+.75)

y=c(seq(-1,1,length.out=500),
  sin(seq(-pi/2,pi/2,length.out=500)),
  seq(-1,1,length.out=500),
  rep(-1,500),
  rep(0,500),
  rep(1,500),
  .5*sin(seq(-pi/2,pi/2,length.out=500))+.5
  ,.5*sin(seq(-pi/2,pi/2,length.out=500))- .5)

z=rbind(x,y)

plot(y~x, xlim=c(-3,3), ylim=c(-3,3))
```



## Create leftMultiply function

Create a function that will left multiply a square matrix against the vector of points.

```
leftMultiply <- function(x,y) {
  x %*% y
}

leftMultiply(matrix(seq(1,9), nrow=3, ncol = 3),diag(3))

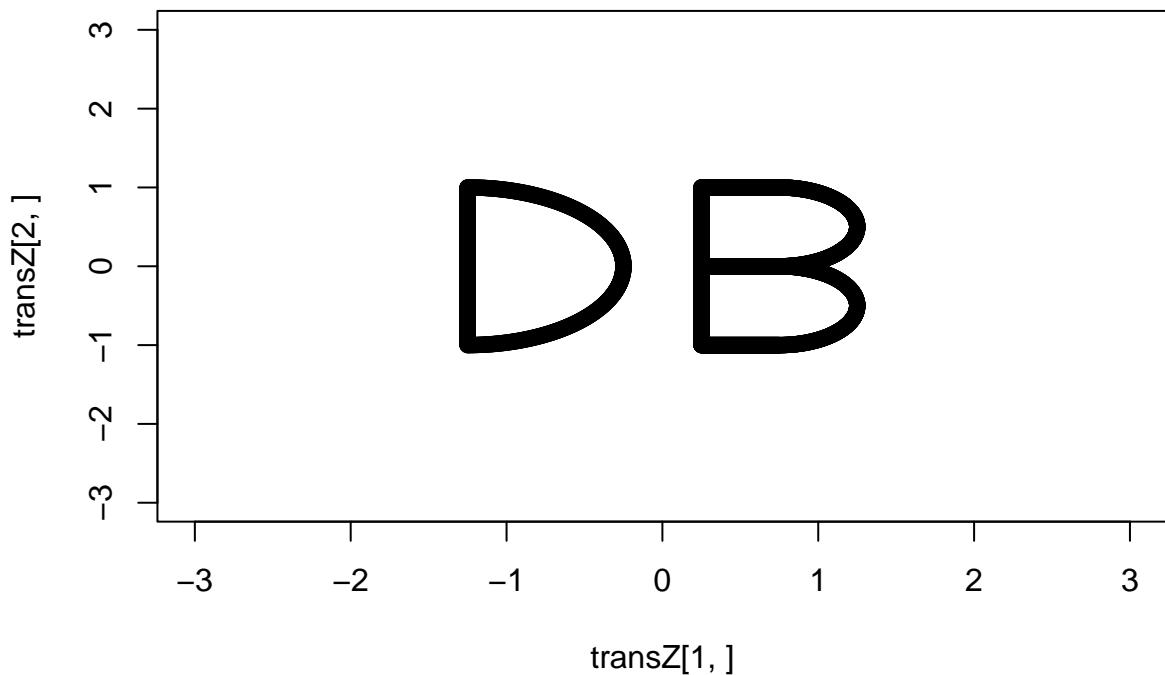
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

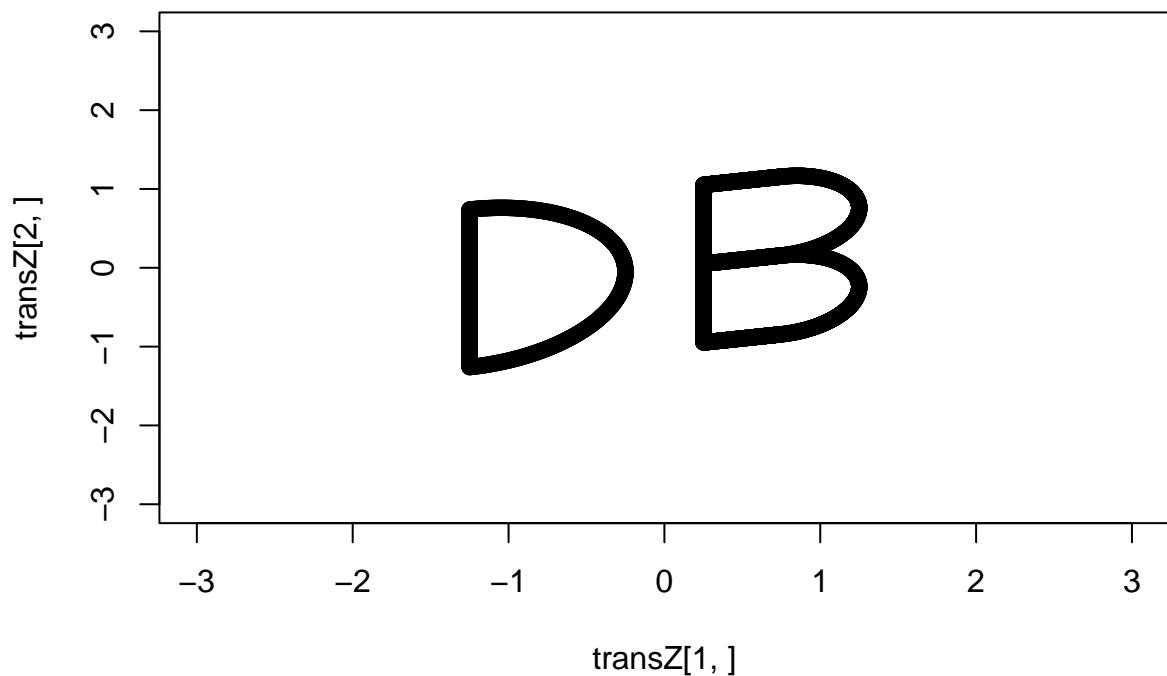
## Shear

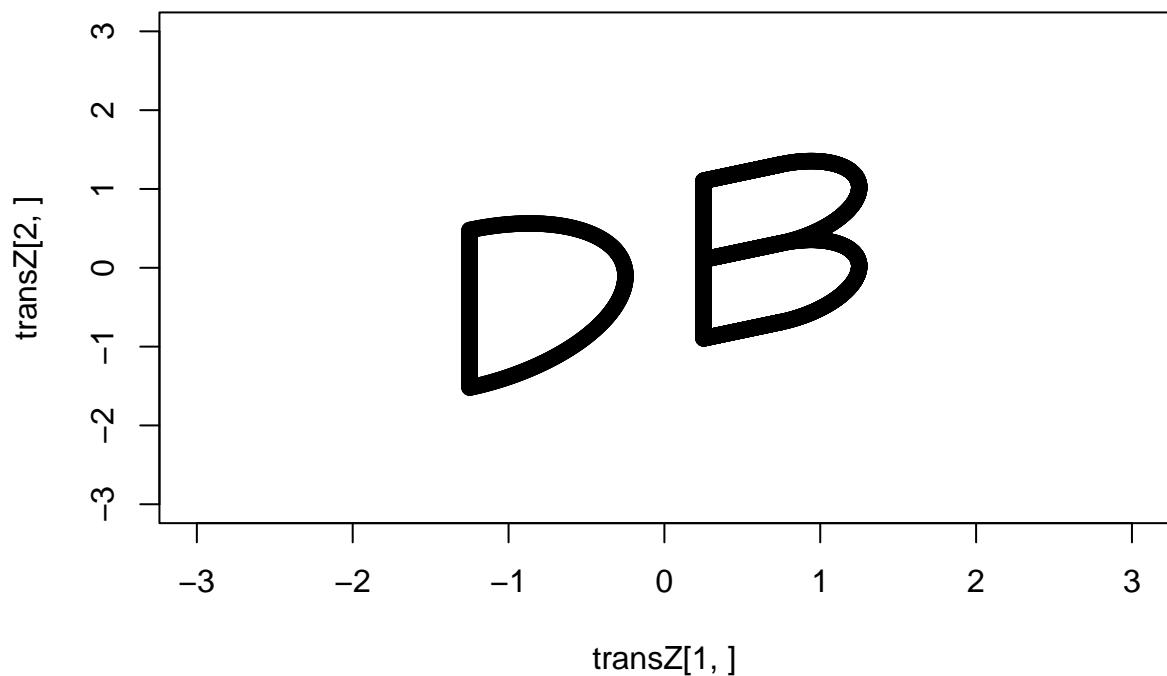
### Horizontal Shear

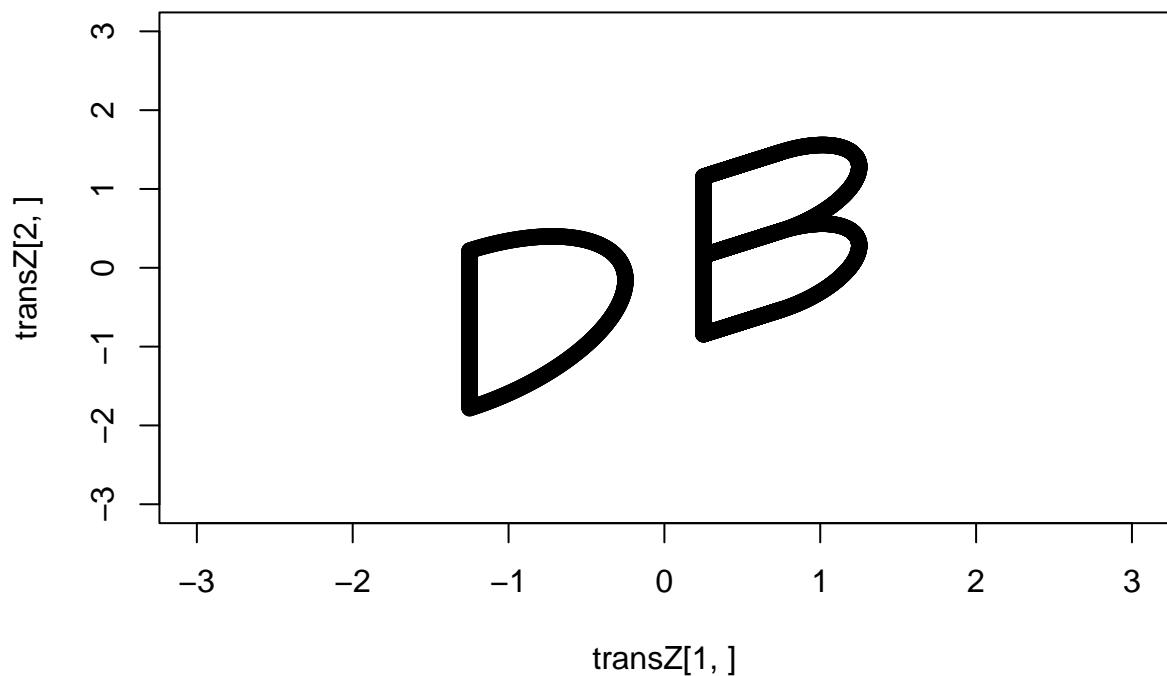
Horizontal shear is a mapping that takes points  $(x,y)$  to  $(x+my,y)$ .

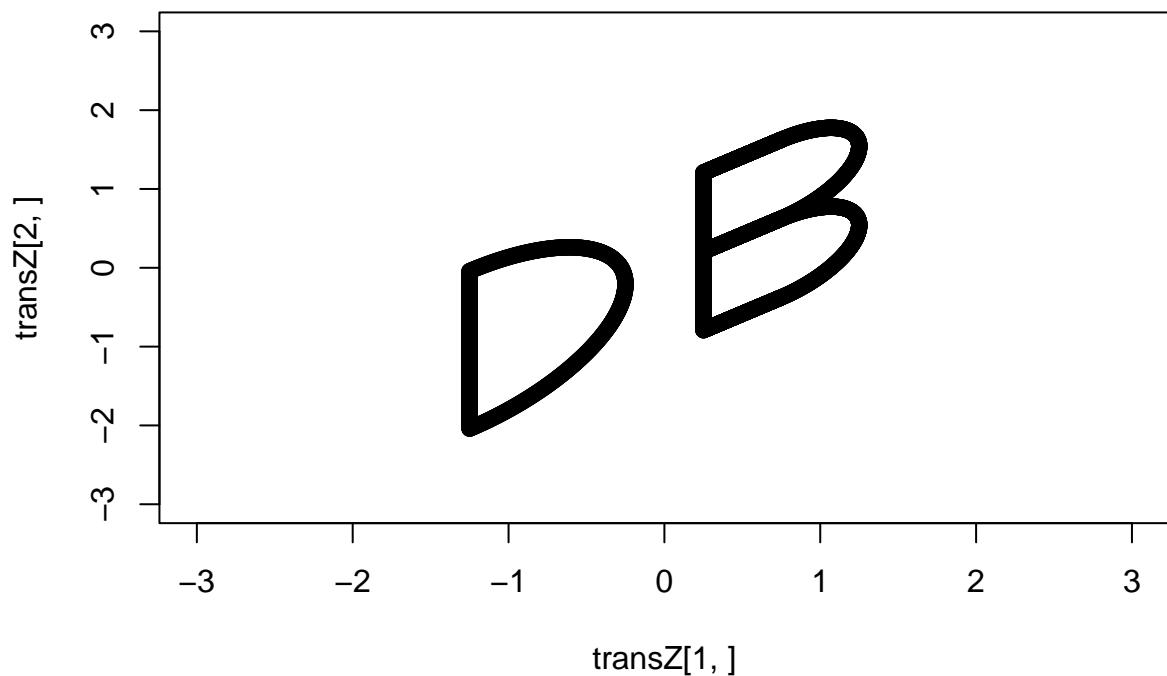
```
x11()  
  
for (shear_factor in seq(0,5,length.out=25)) {  
  transZ <- apply(z,2,function(x) leftMultiply(x,matrix(c(1,0,shear_factor,1),nrow = 2, ncol = 2)))  
  plot(transZ[2,]~transZ[1,], xlim = c(-3,3), ylim = c(-3,3))  
}
```

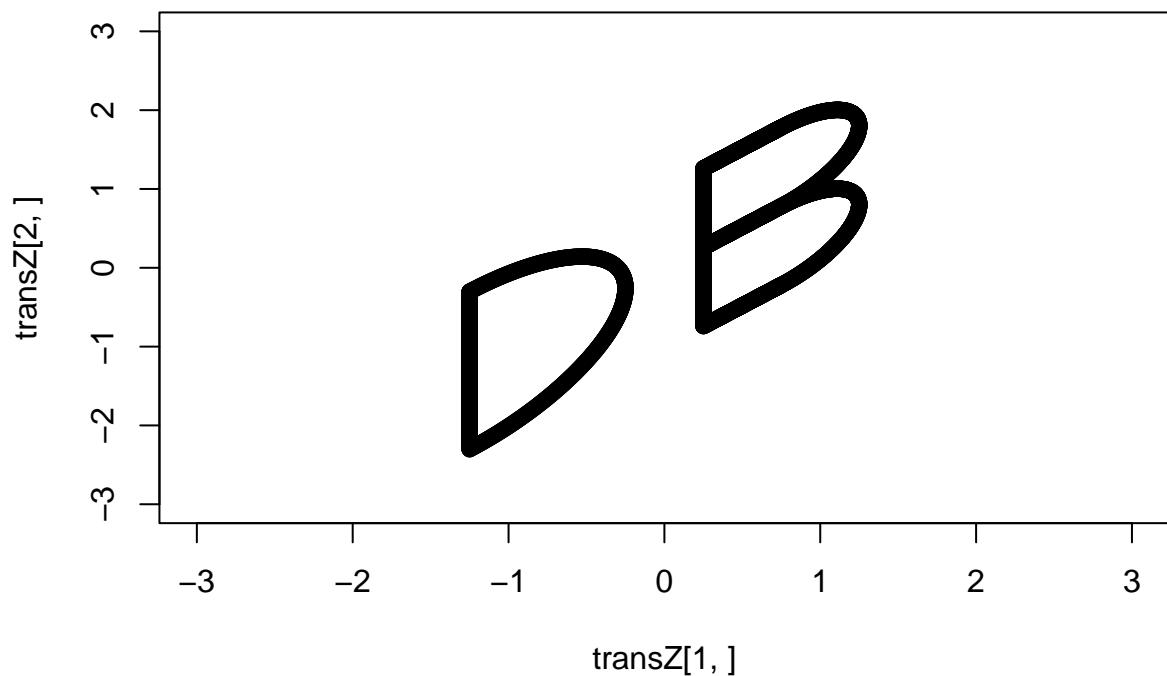


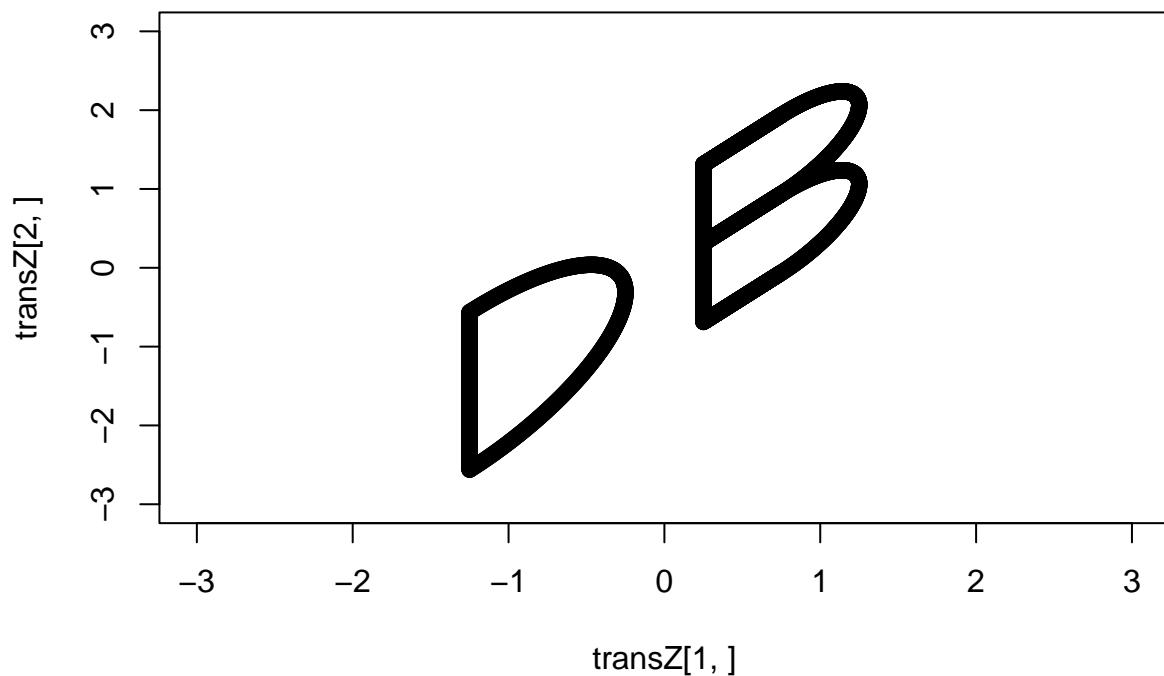


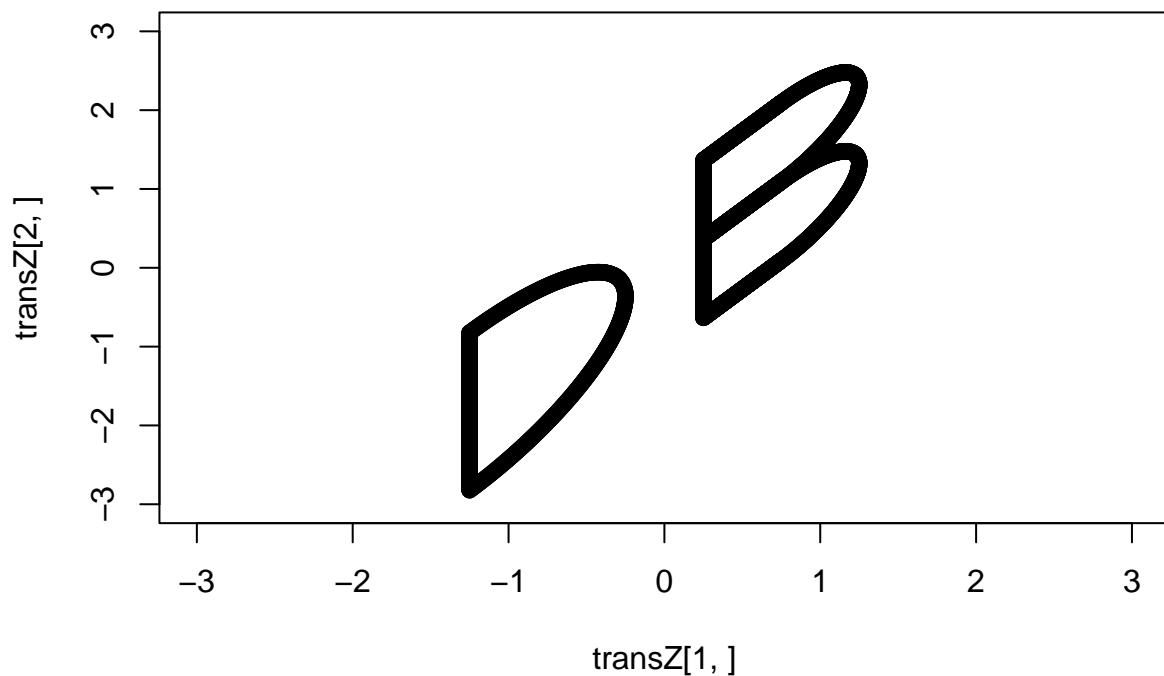


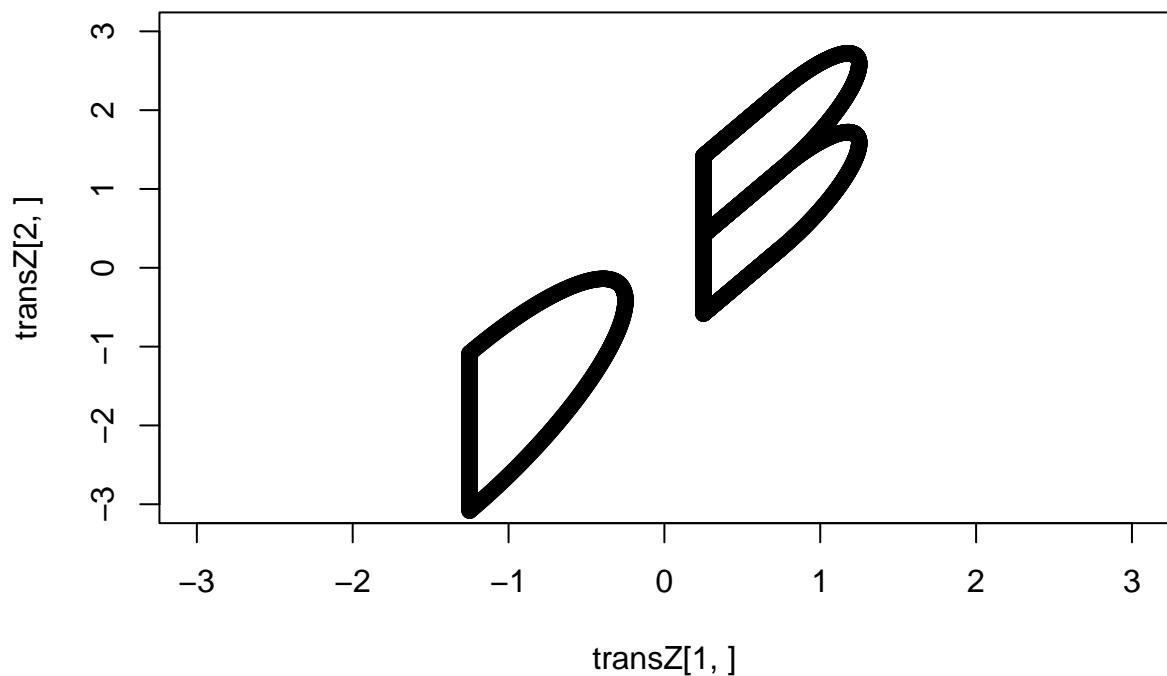


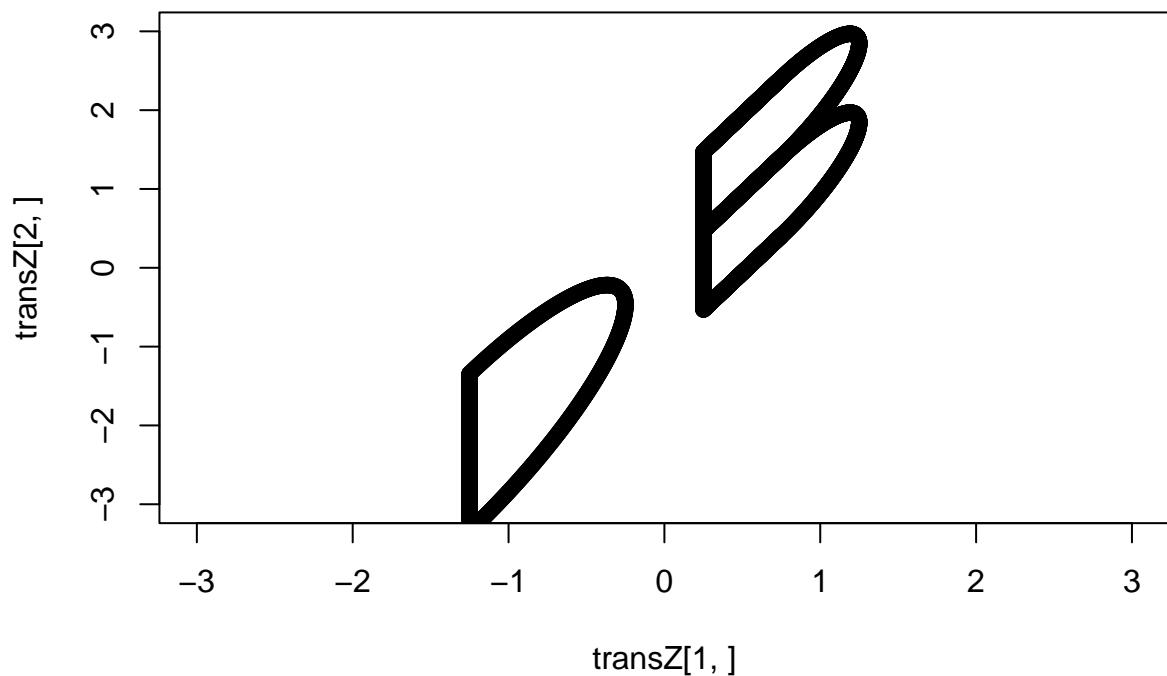


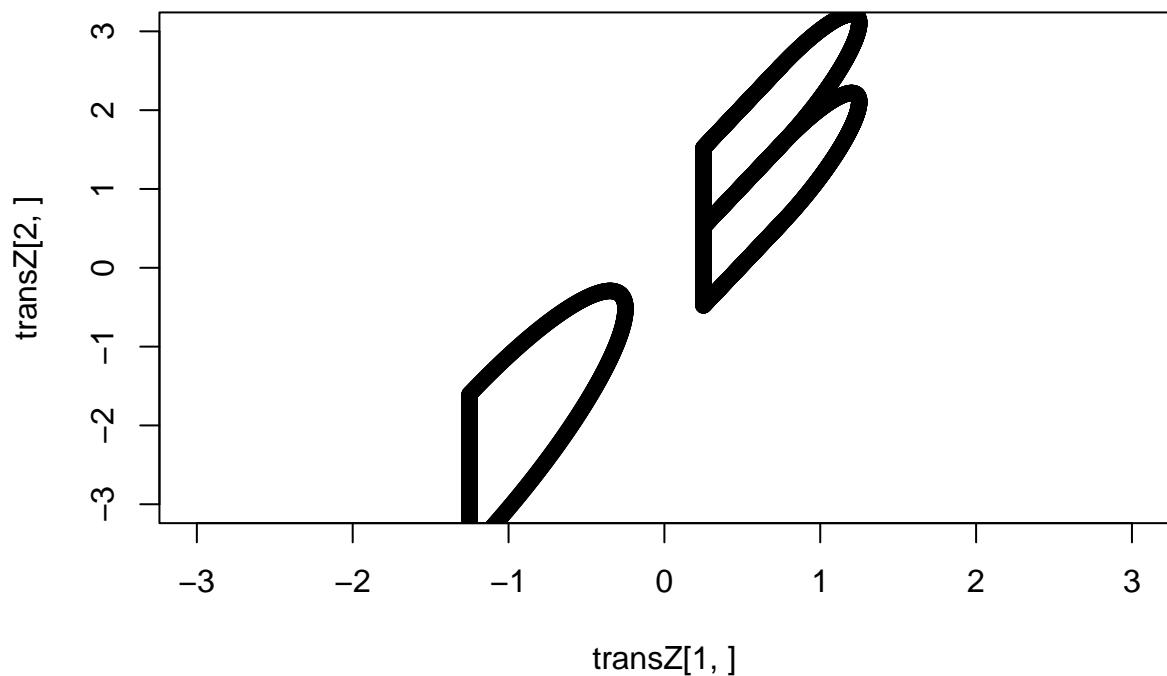


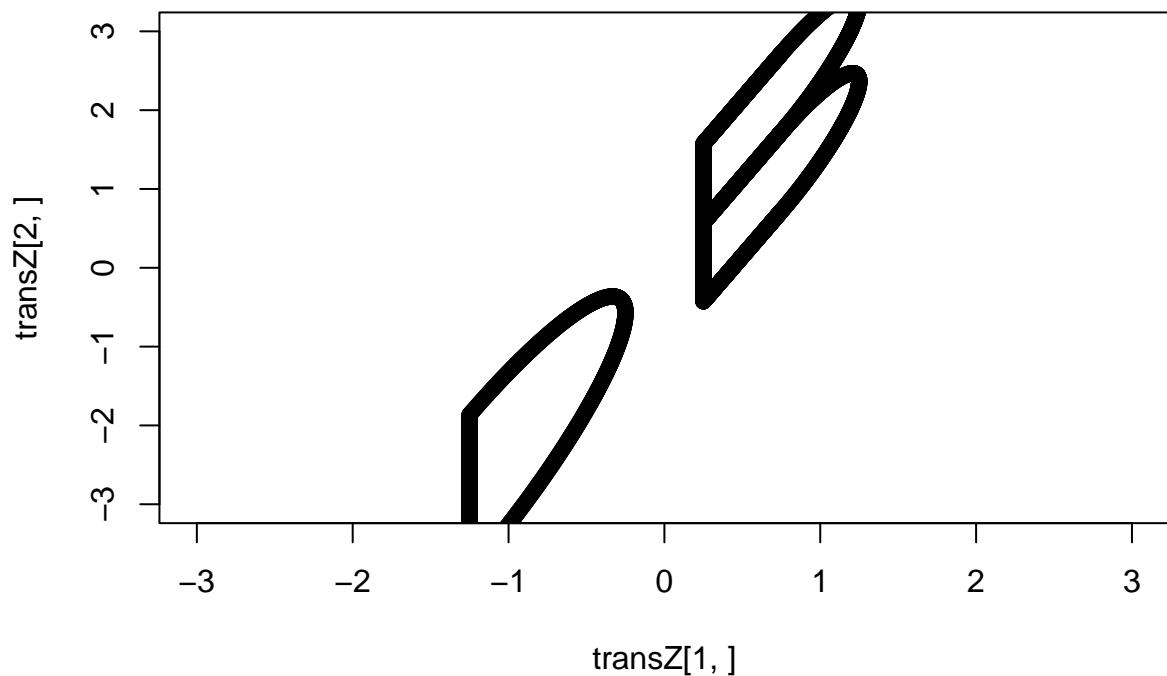


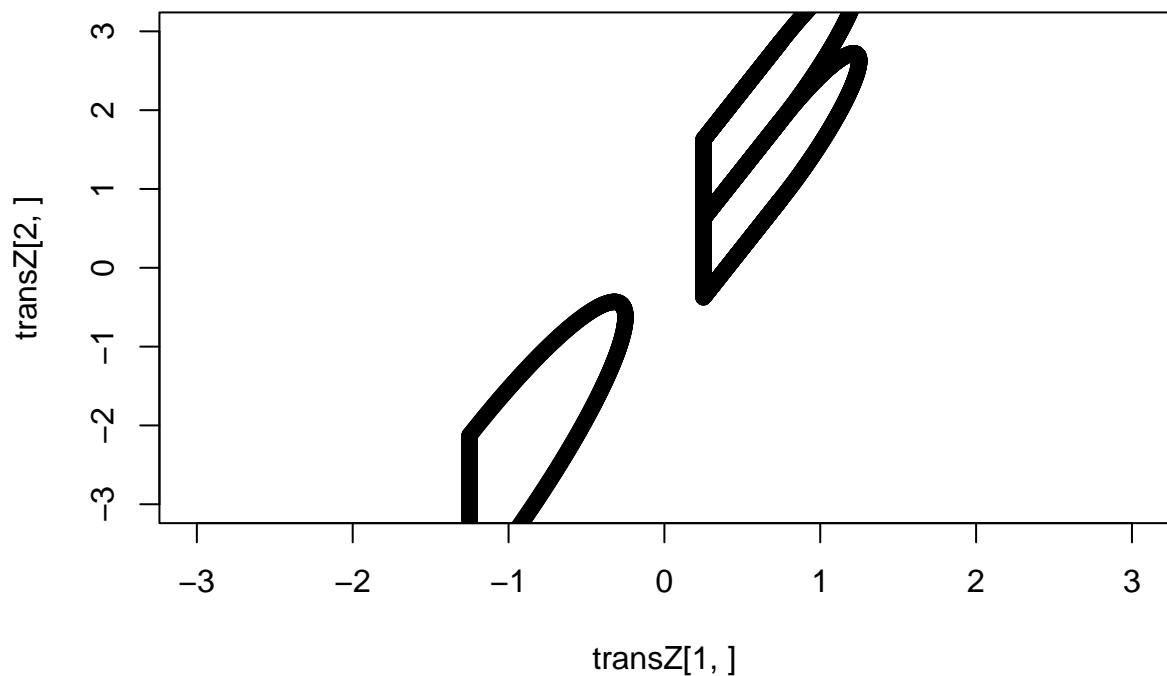


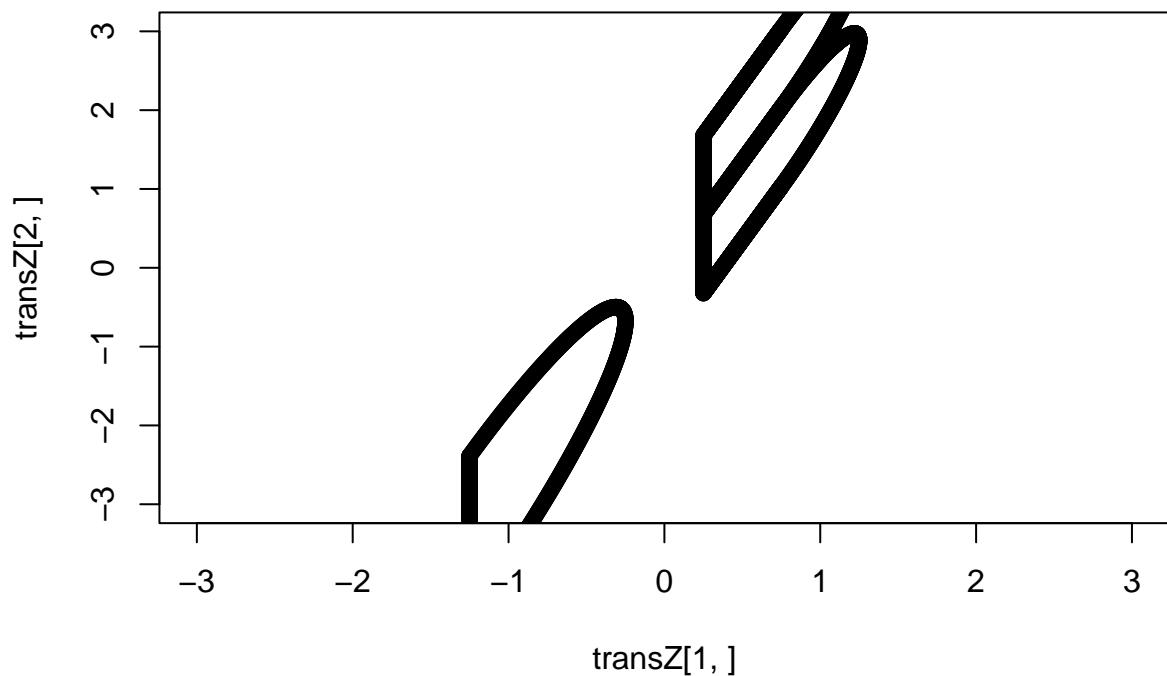


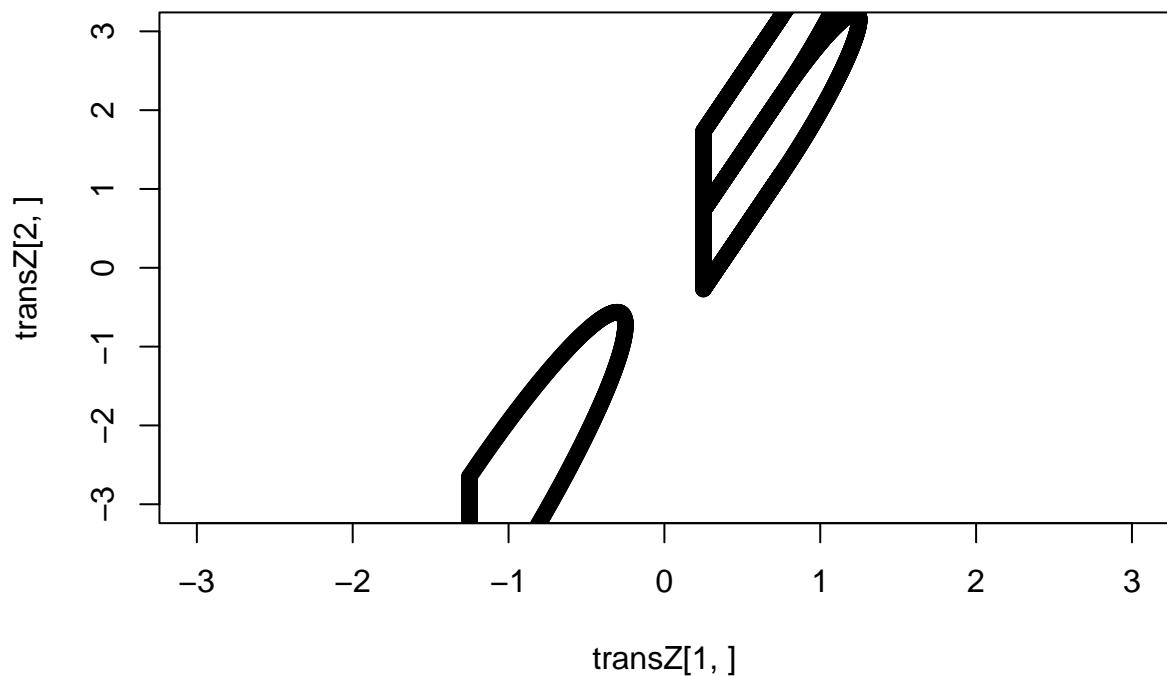


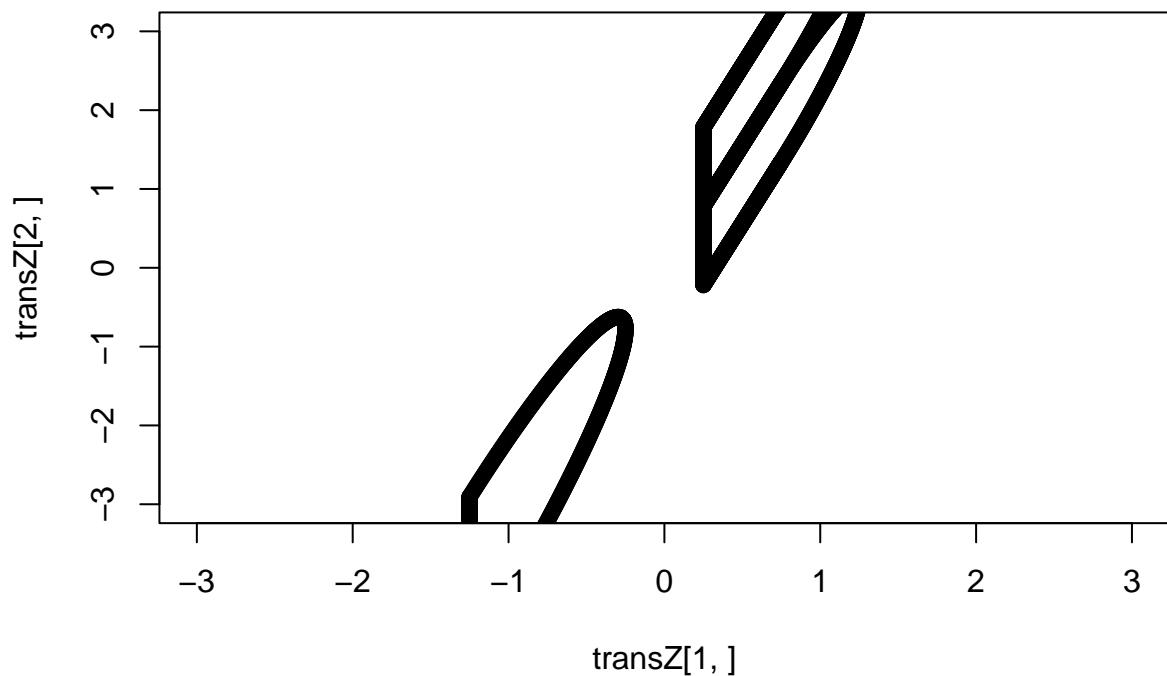


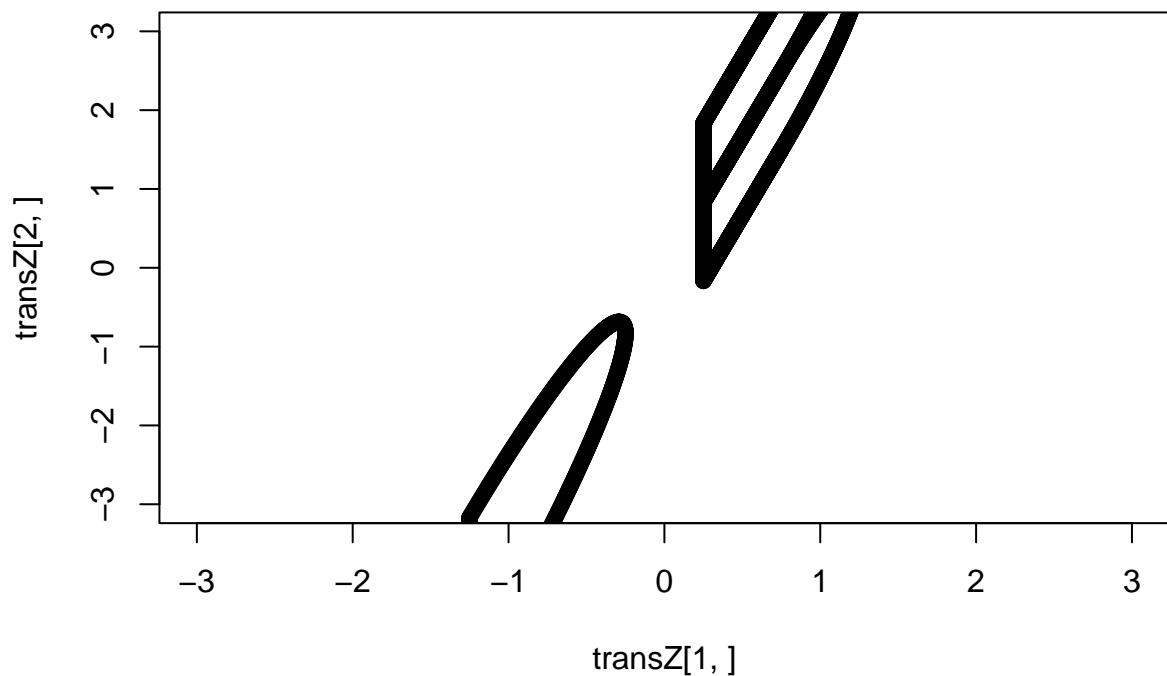


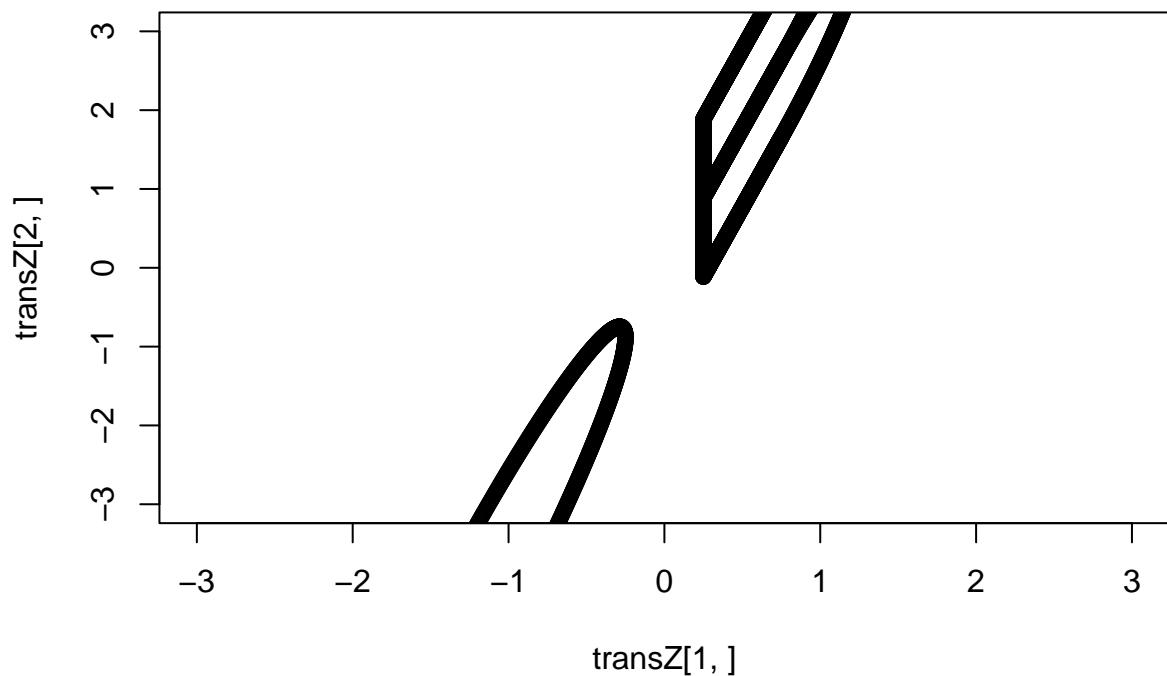


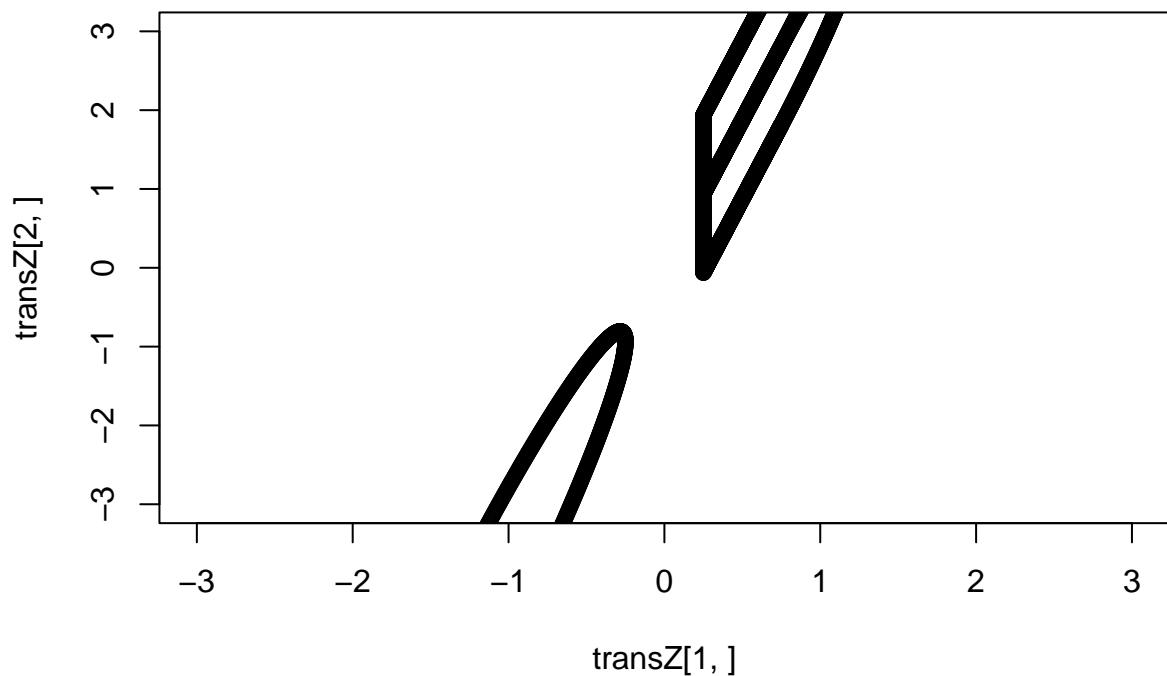


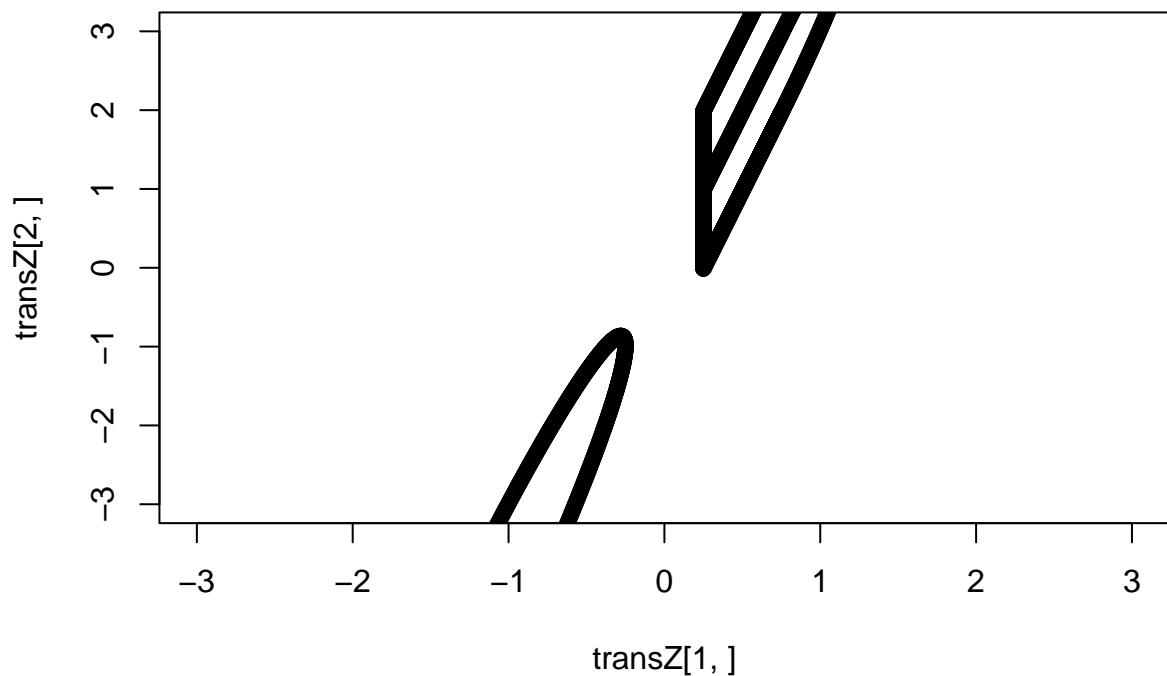


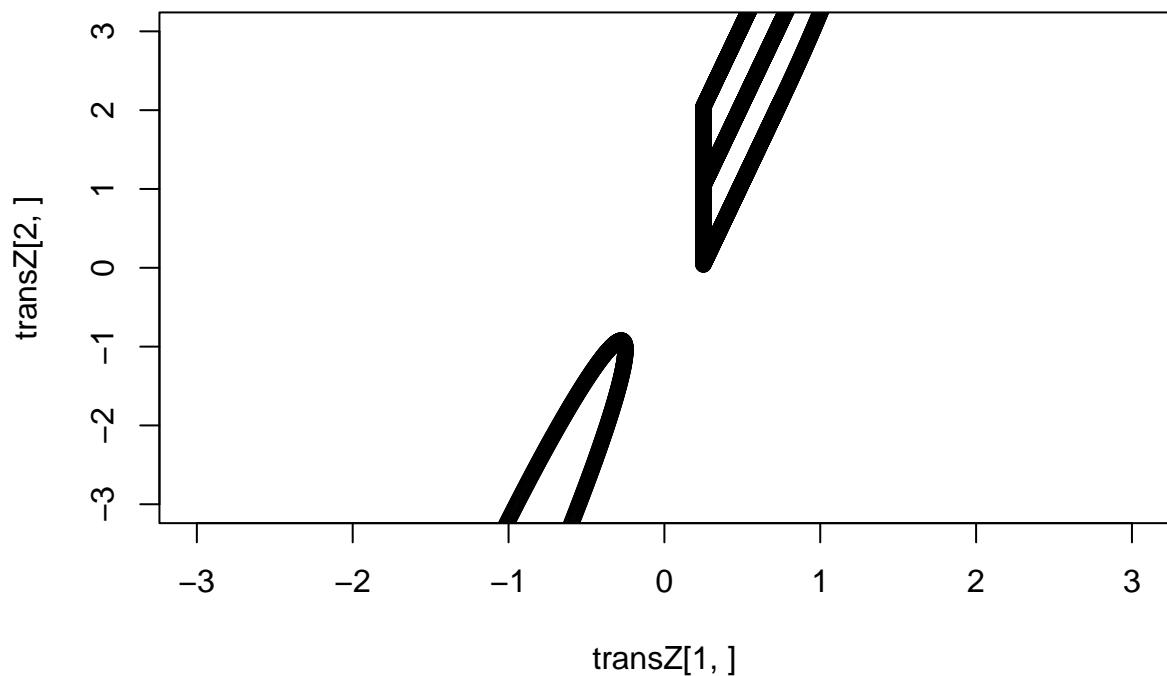


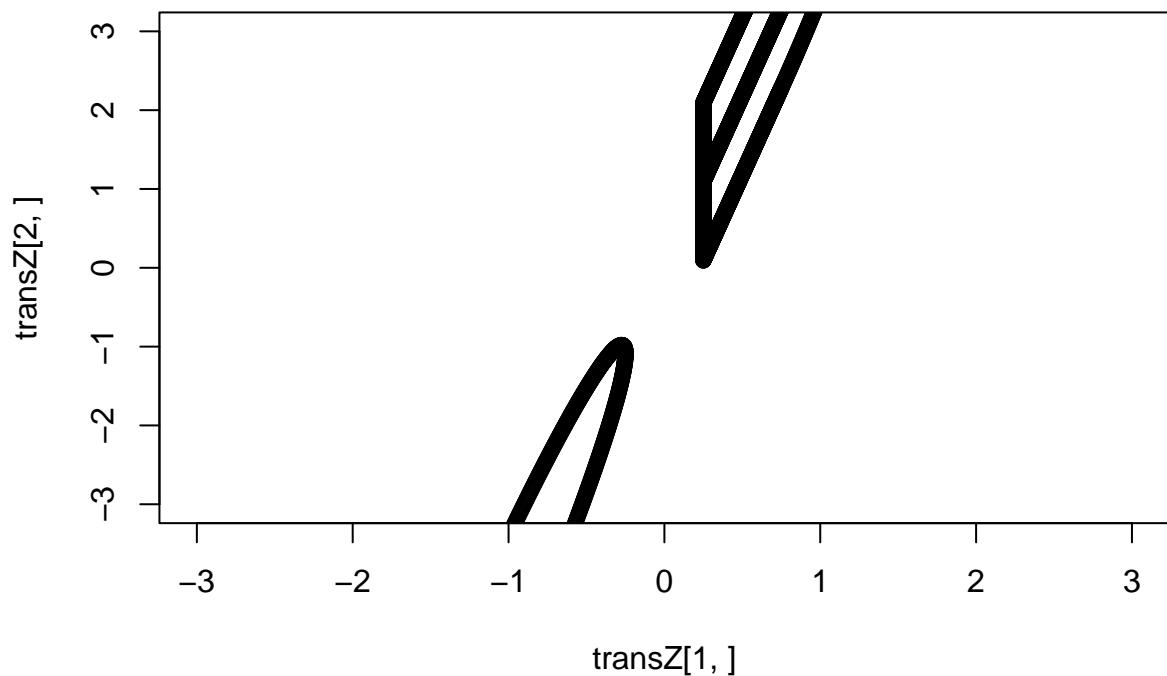


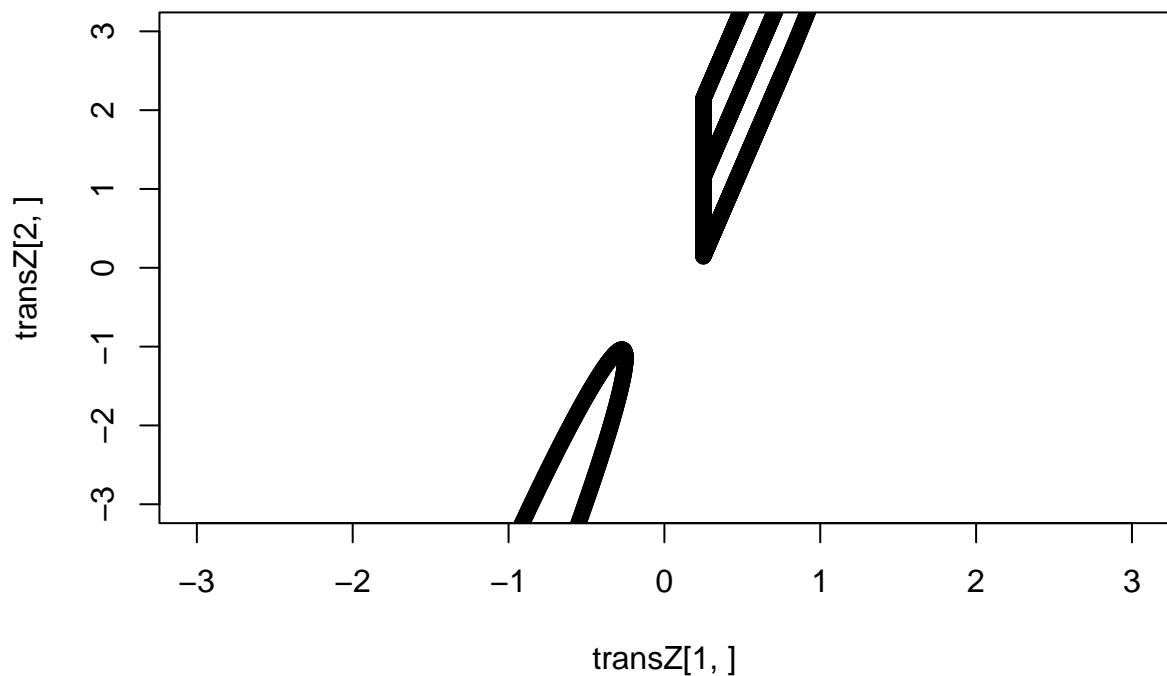


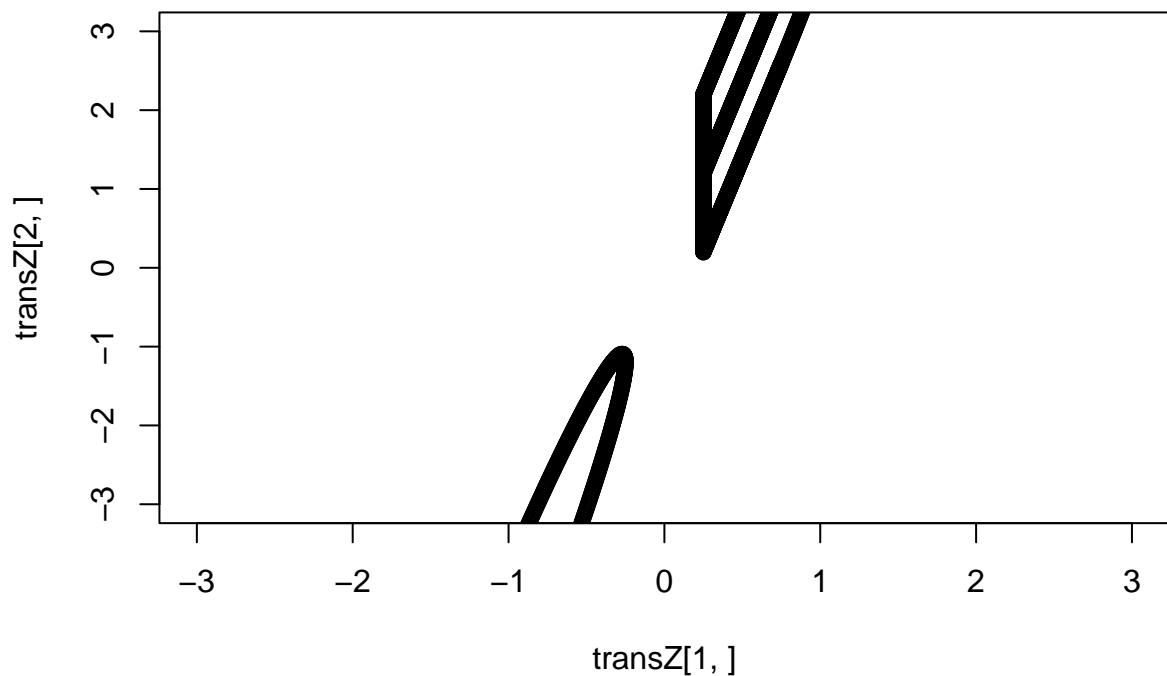


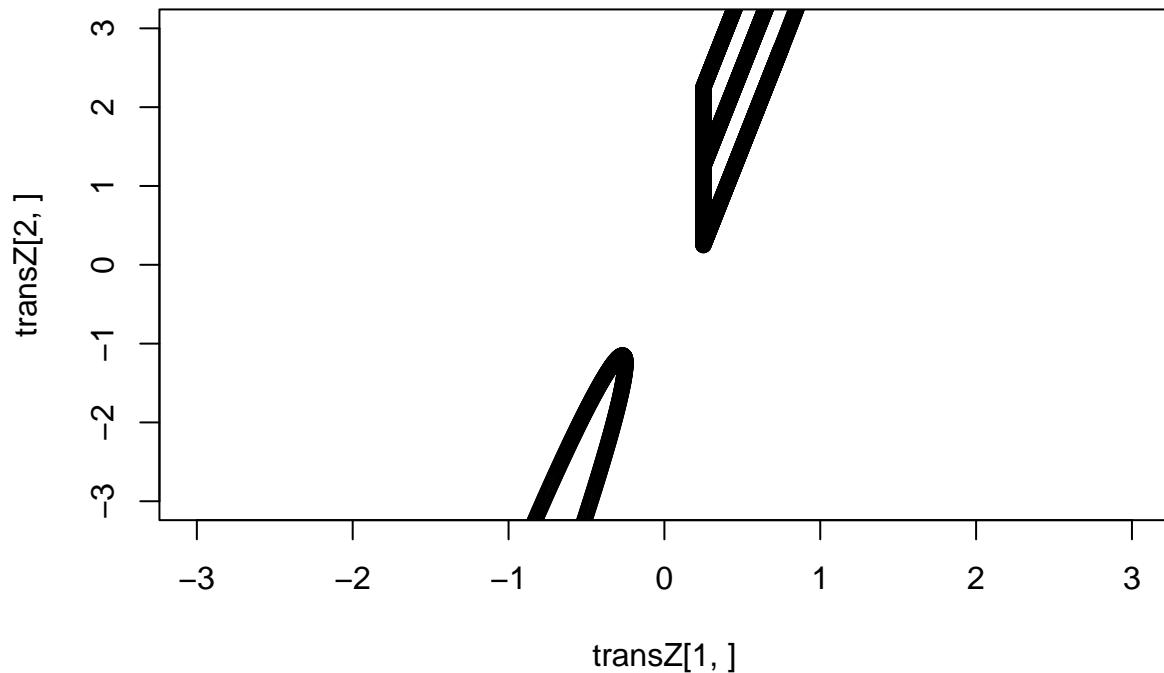








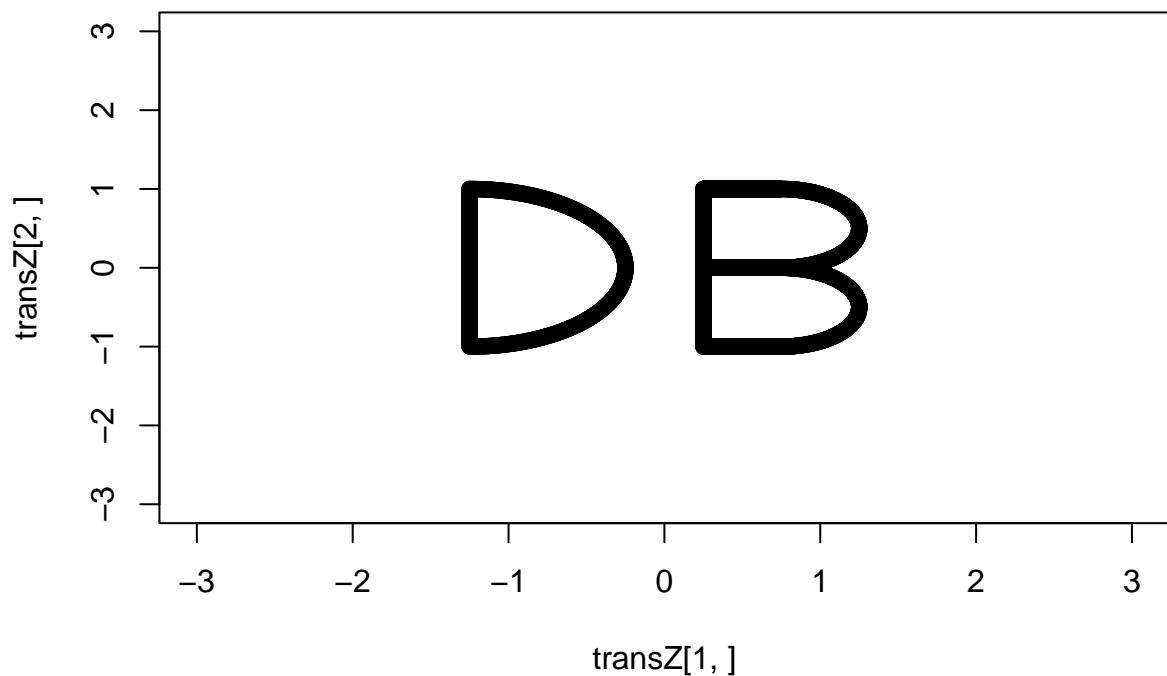


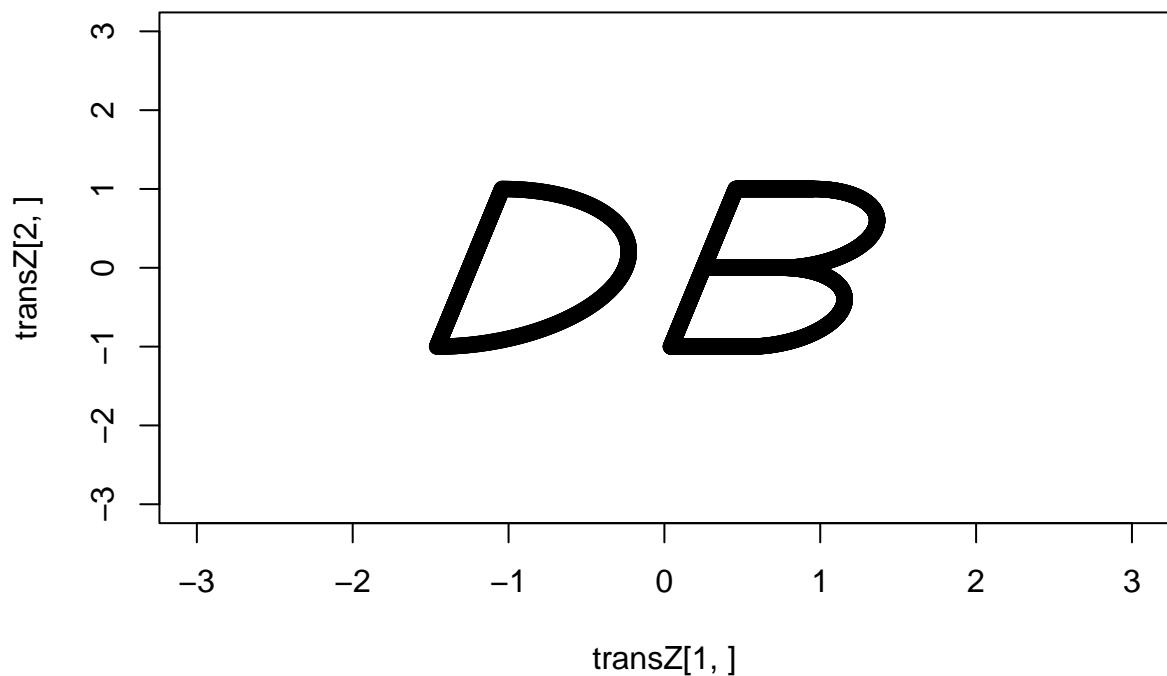


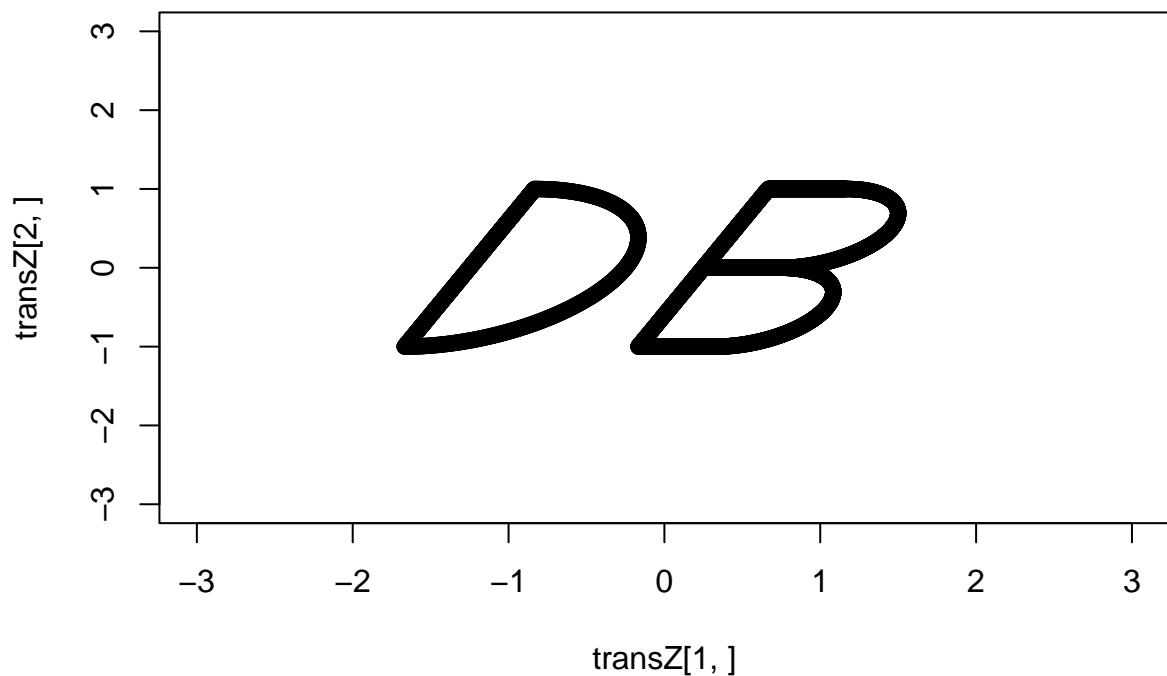
## Vertical Shear

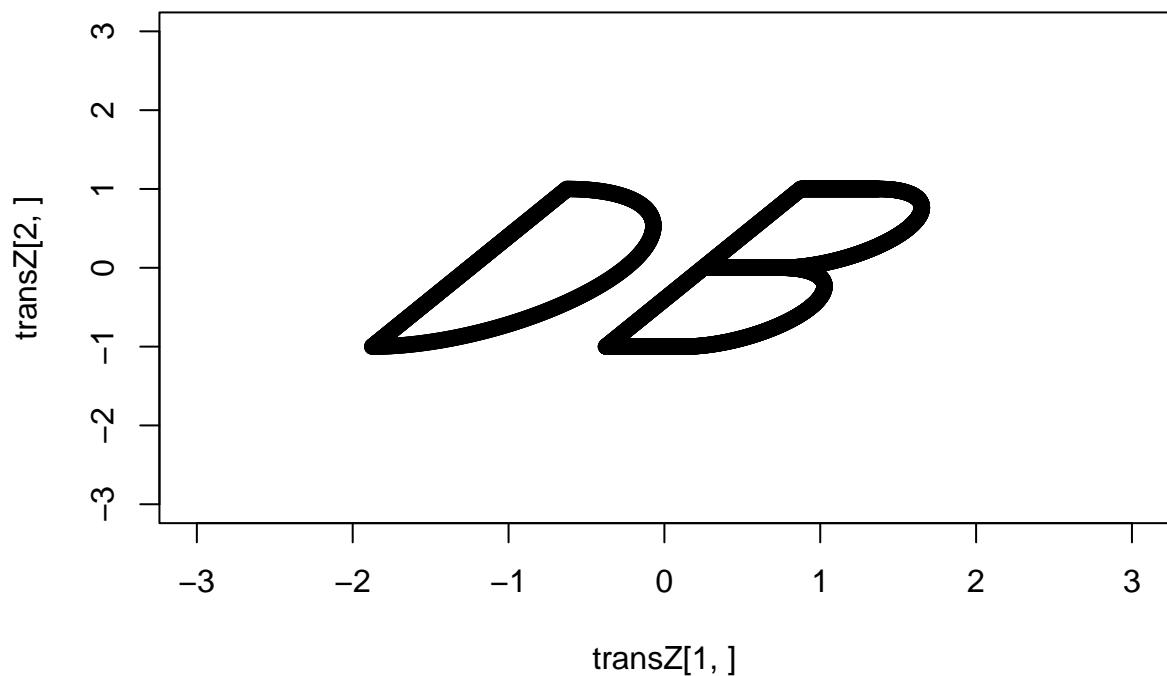
Vertical shear is a mapping that takes points  $(x,y)$  to  $(x, mx+y)$ .

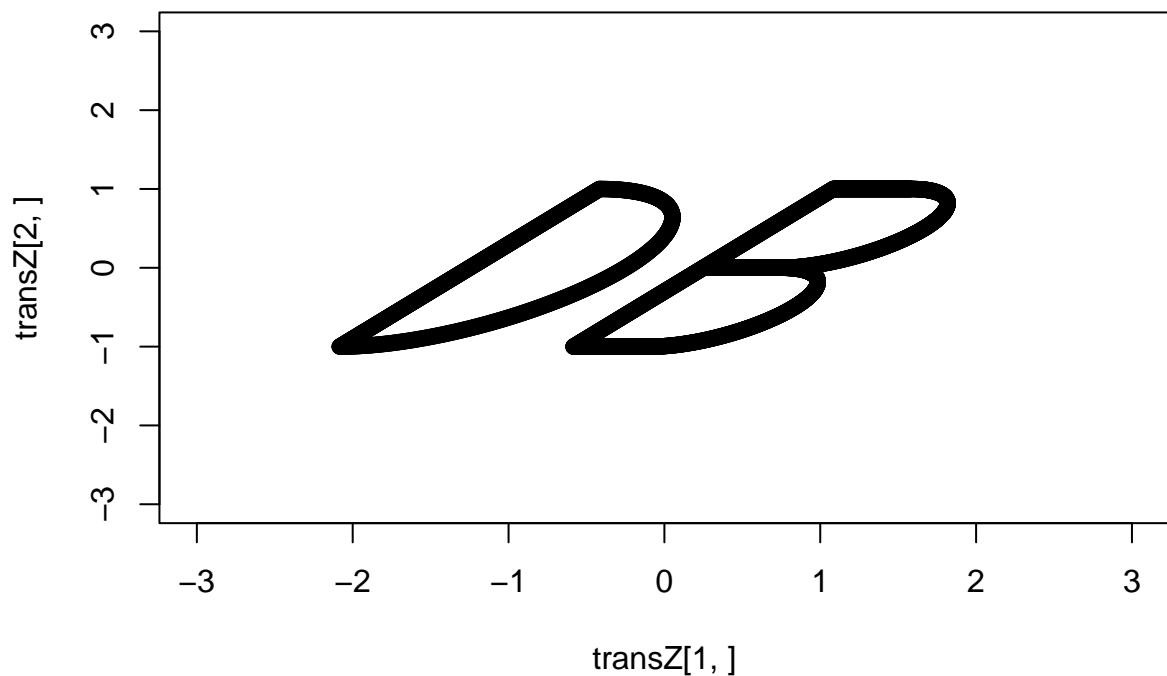
```
x11()
for (shear_factor in seq(0,5,length.out=25)) {
  transZ <- apply(z,2,function(x) leftMultiply(x,matrix(c(1,shear_factor,0,1),nrow = 2, ncol = 2)))
  plot(transZ[2,]-transZ[1,], xlim = c(-3,3), ylim = c(-3,3))
}
```



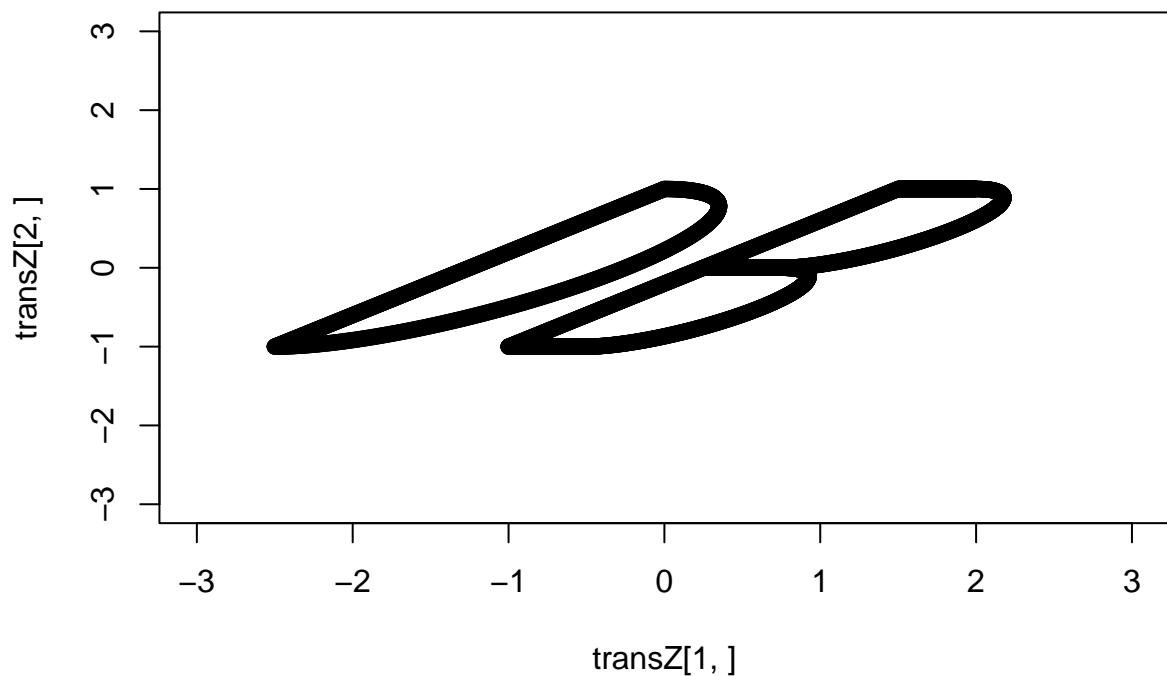


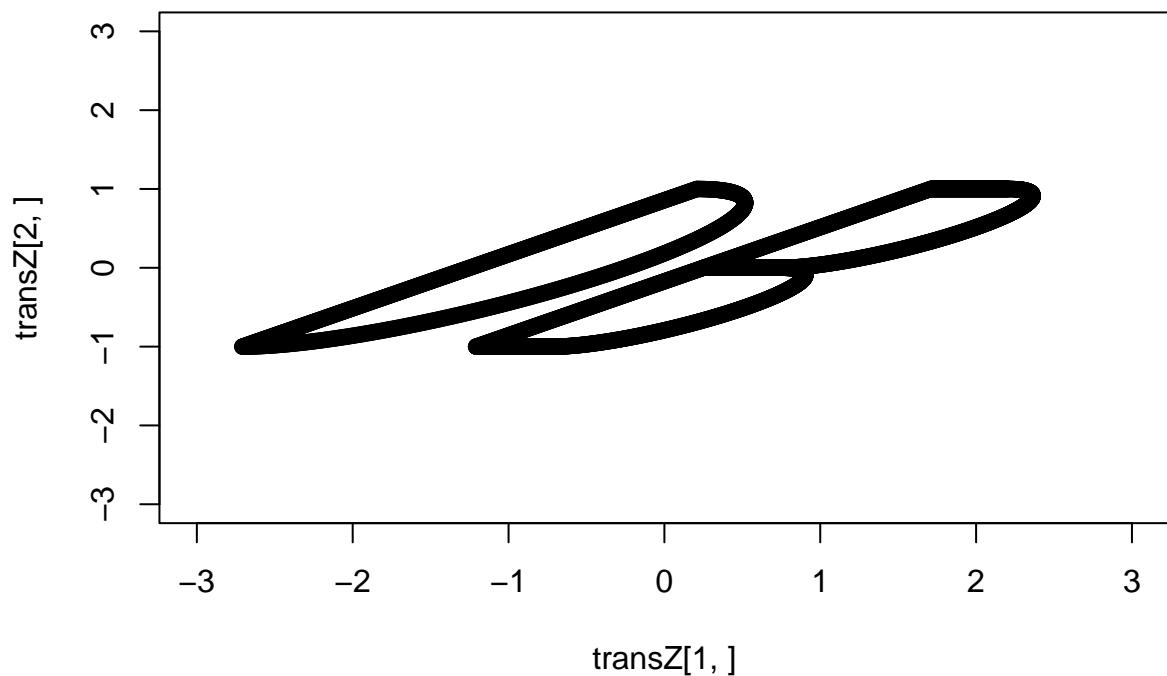


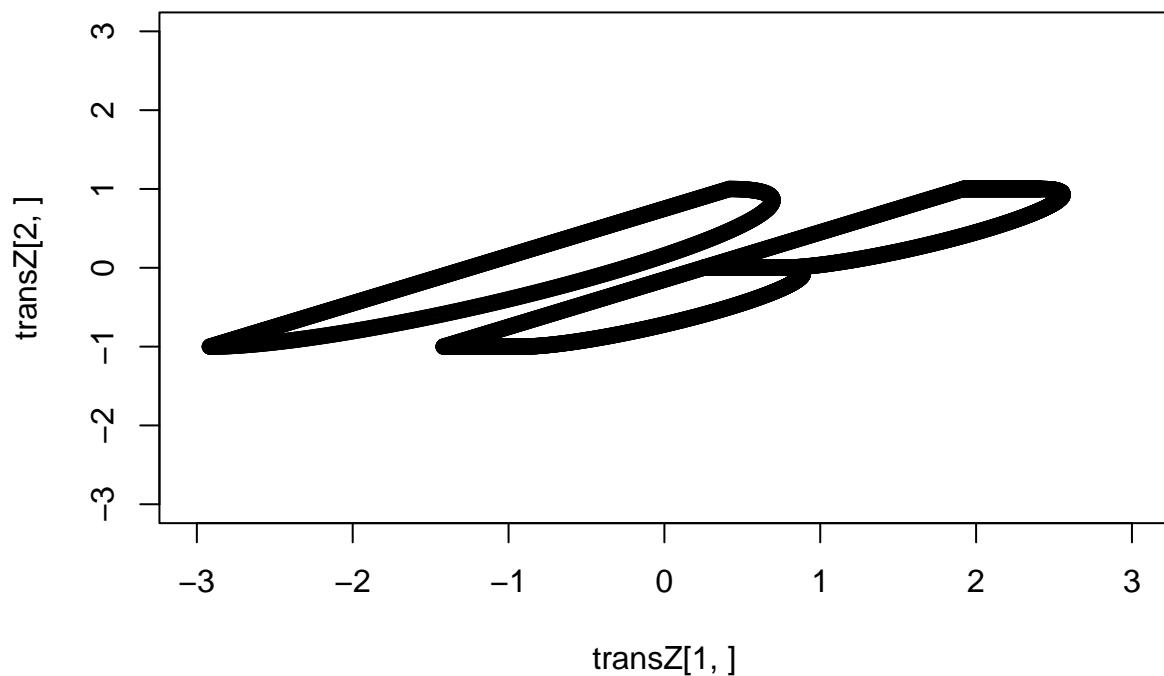


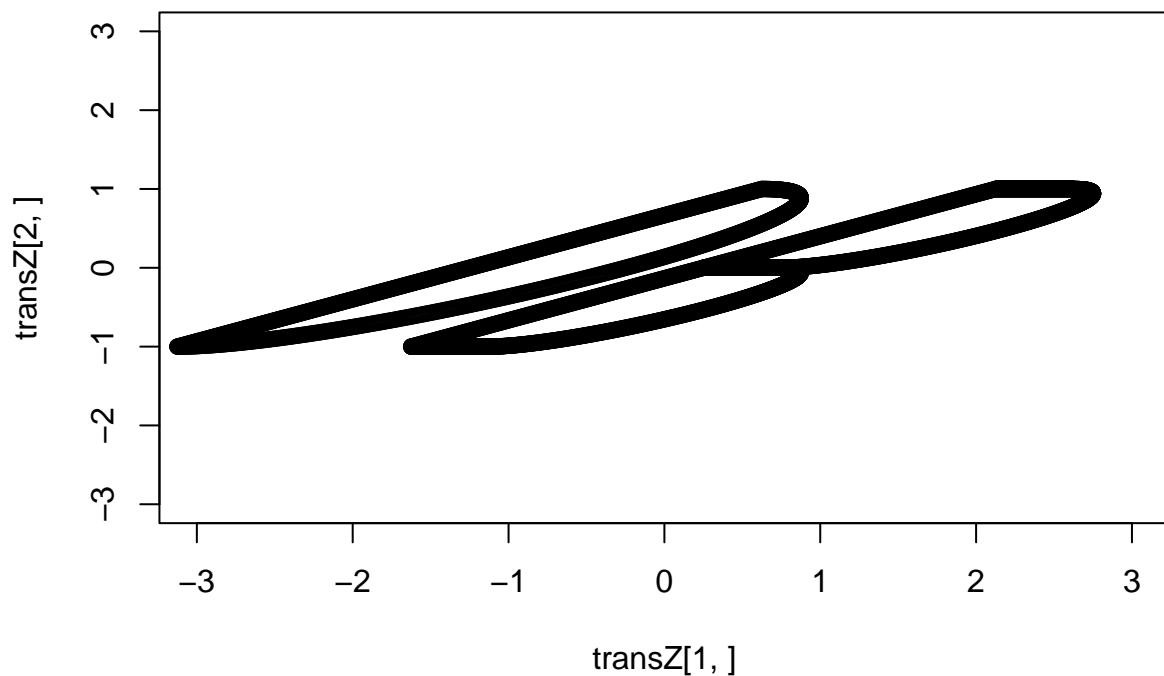


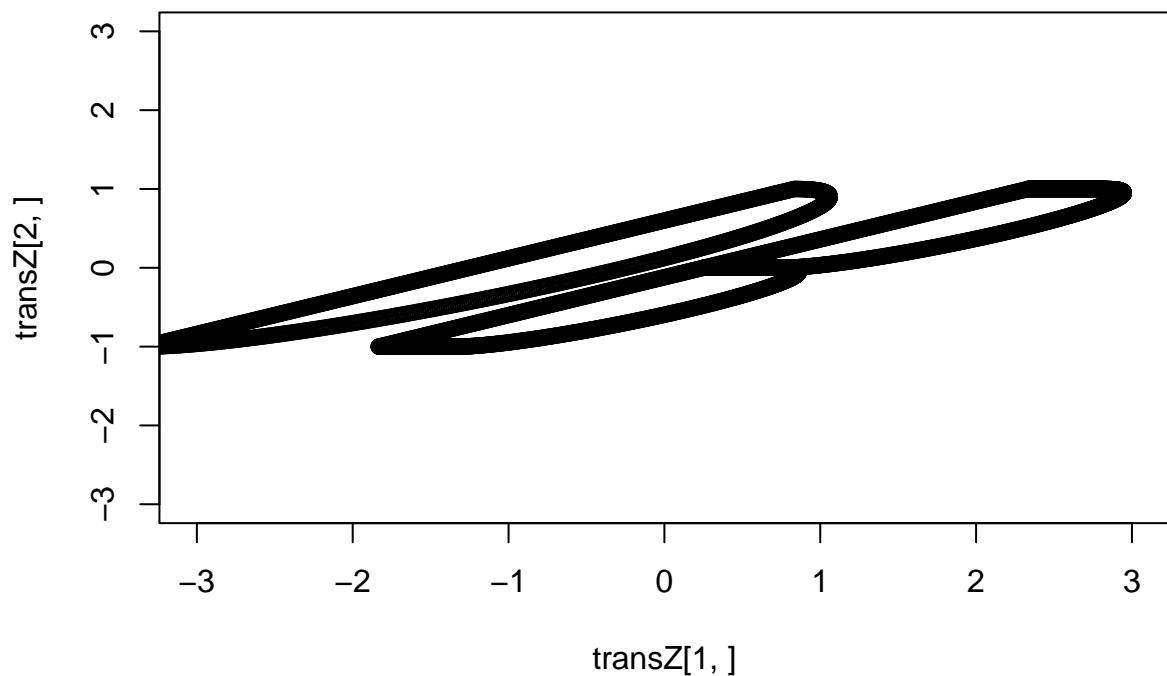


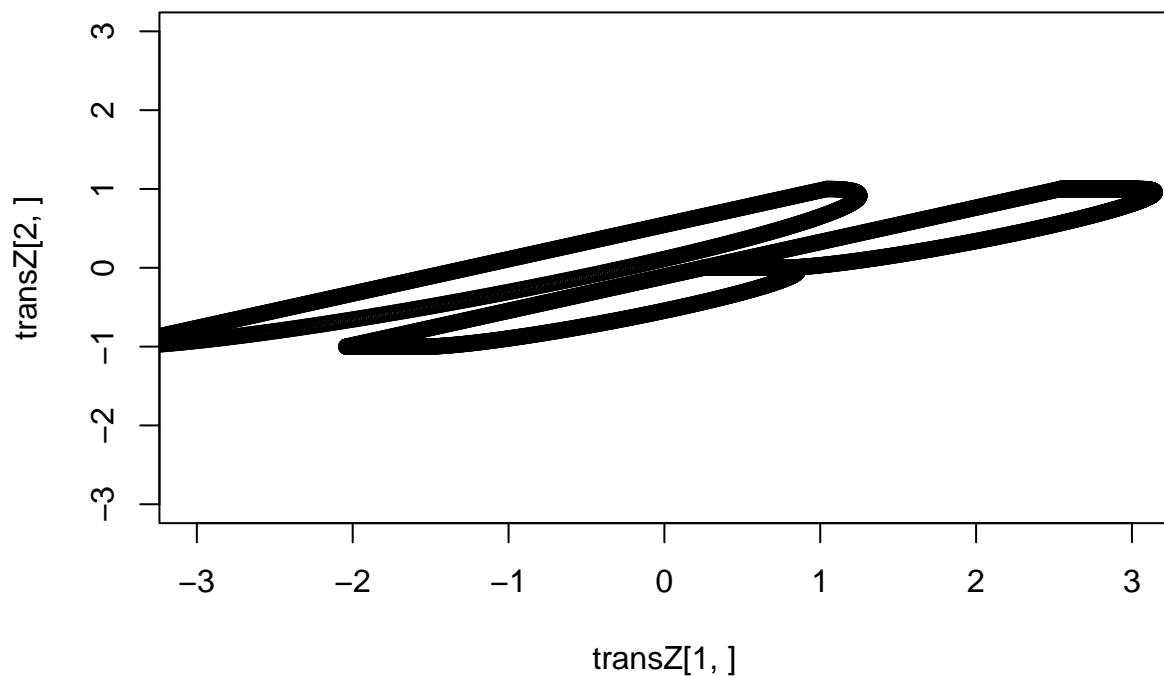


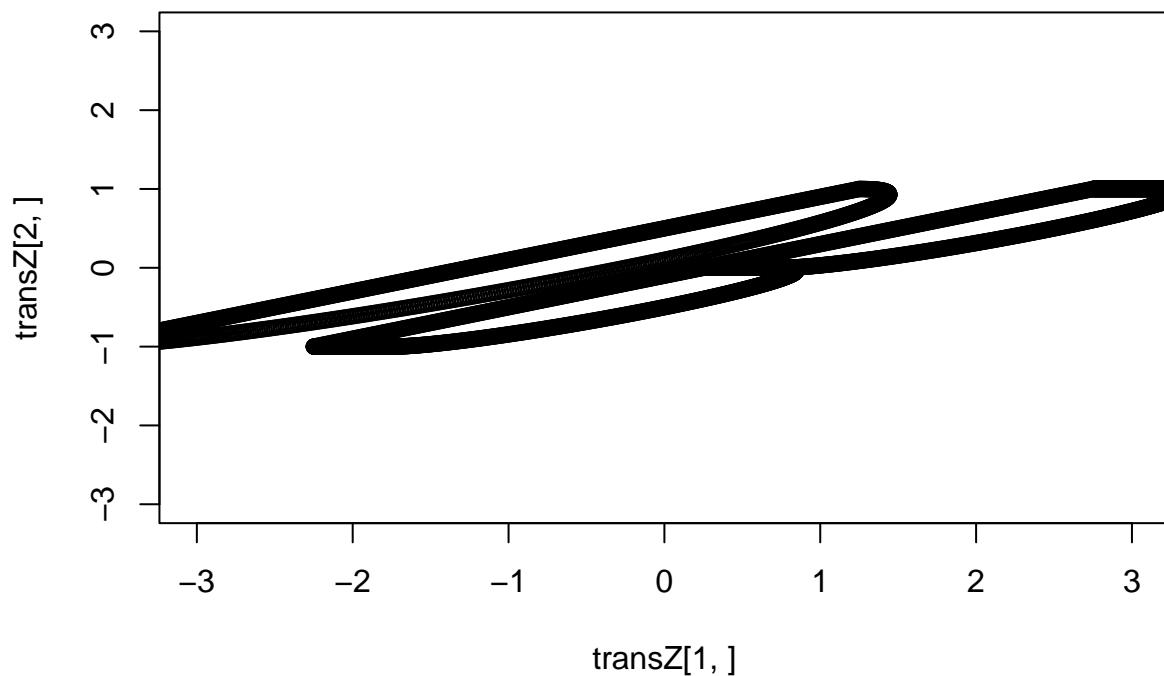


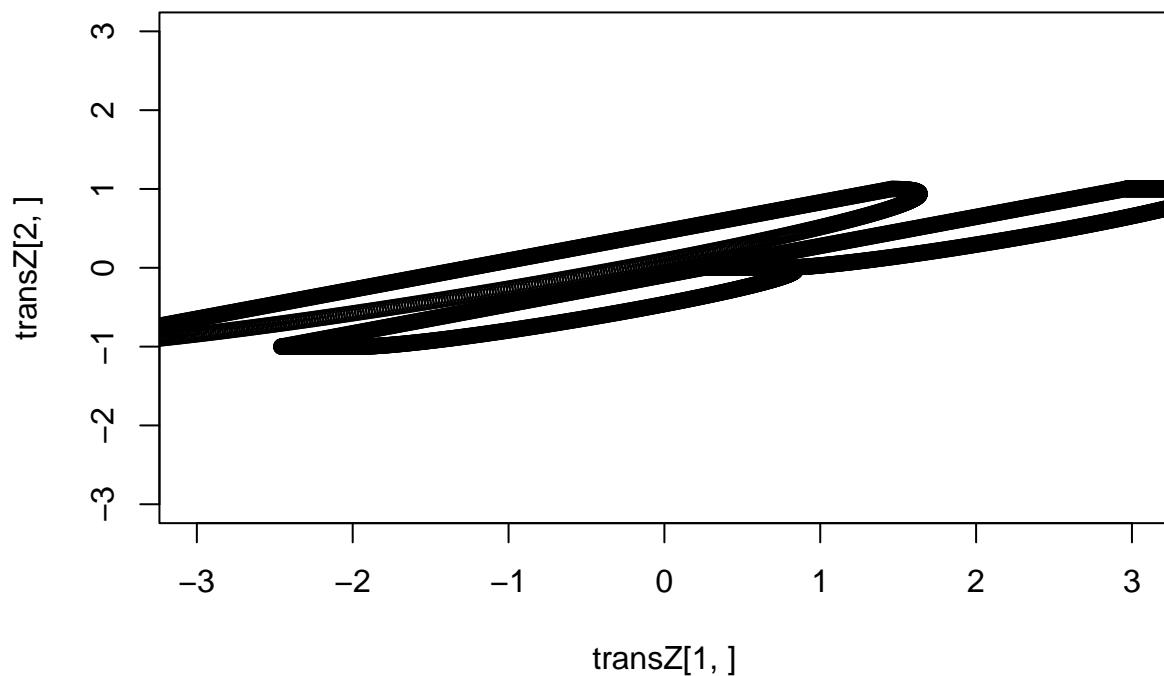


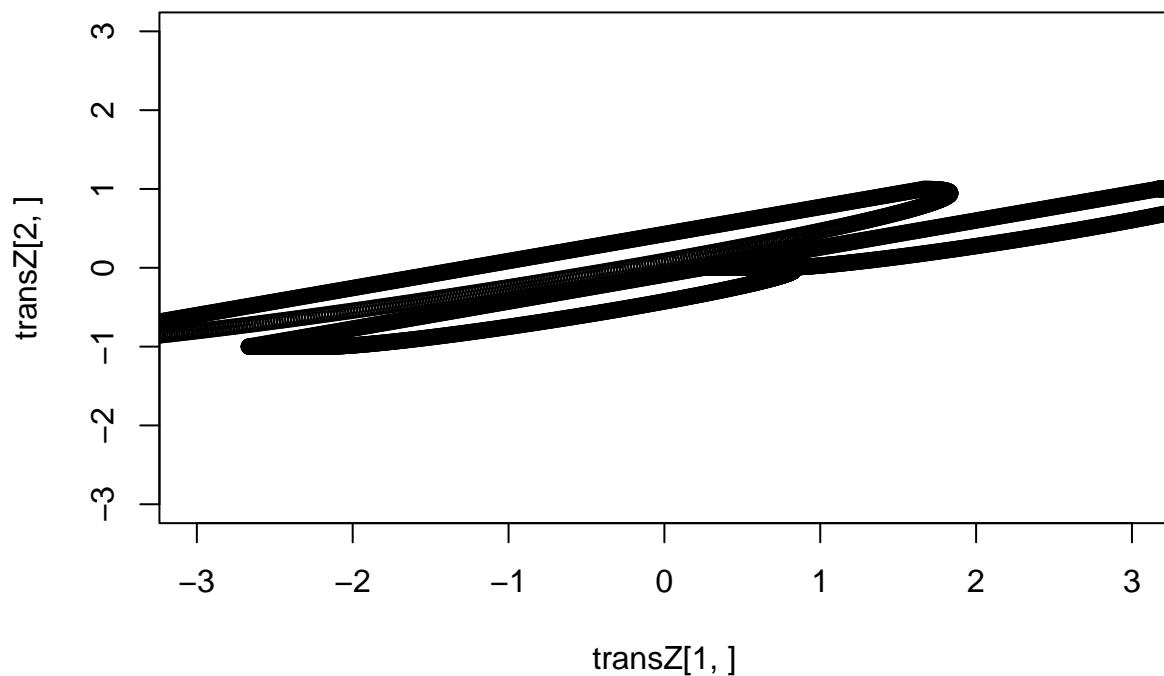


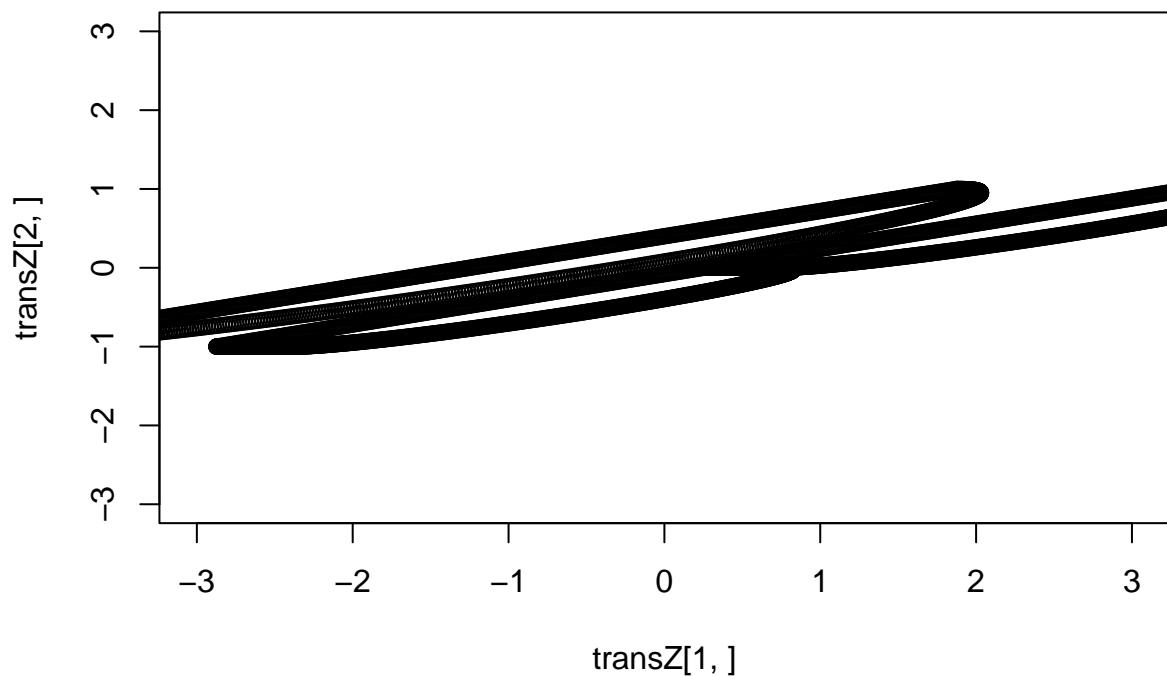


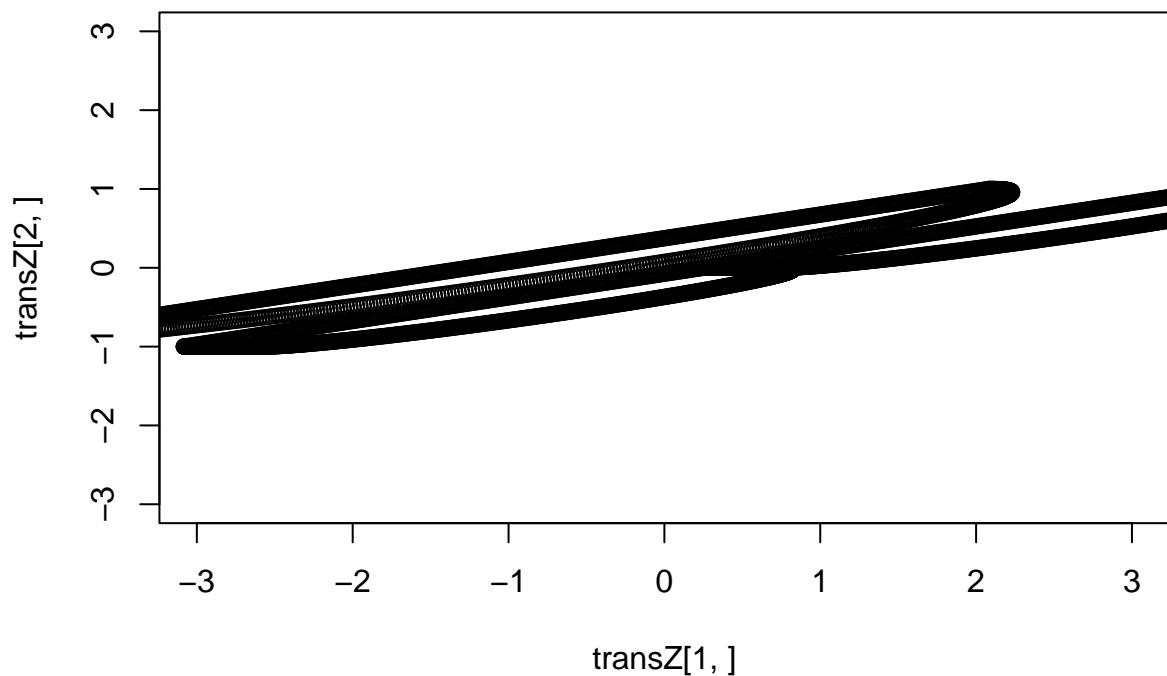


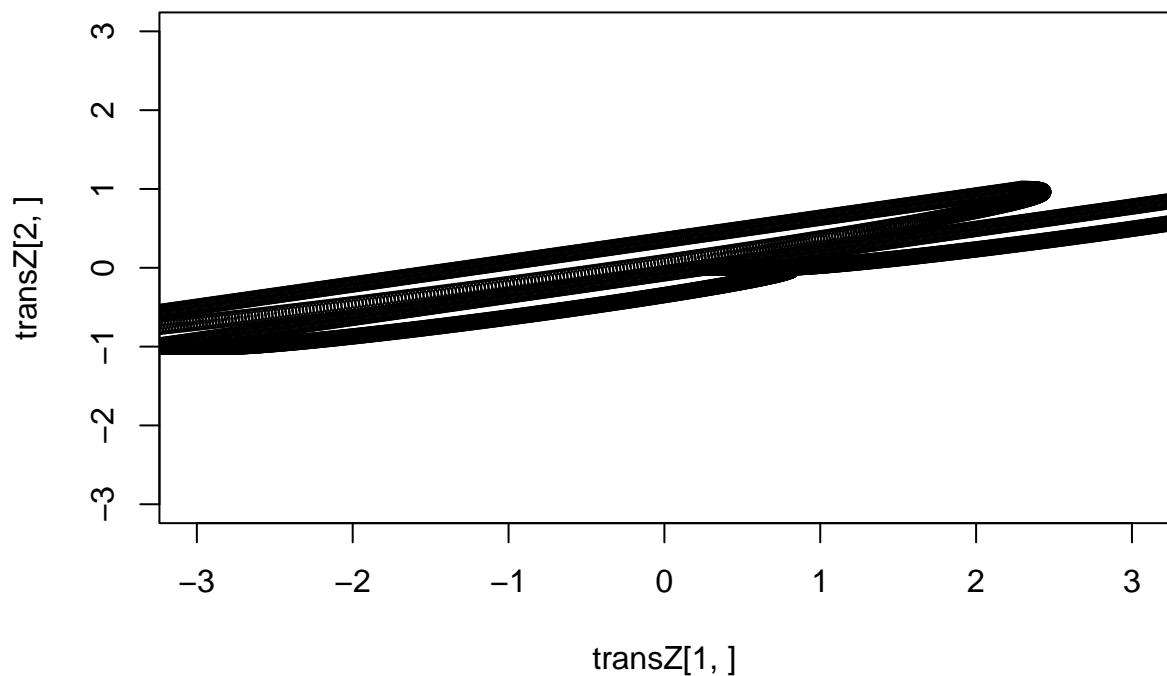


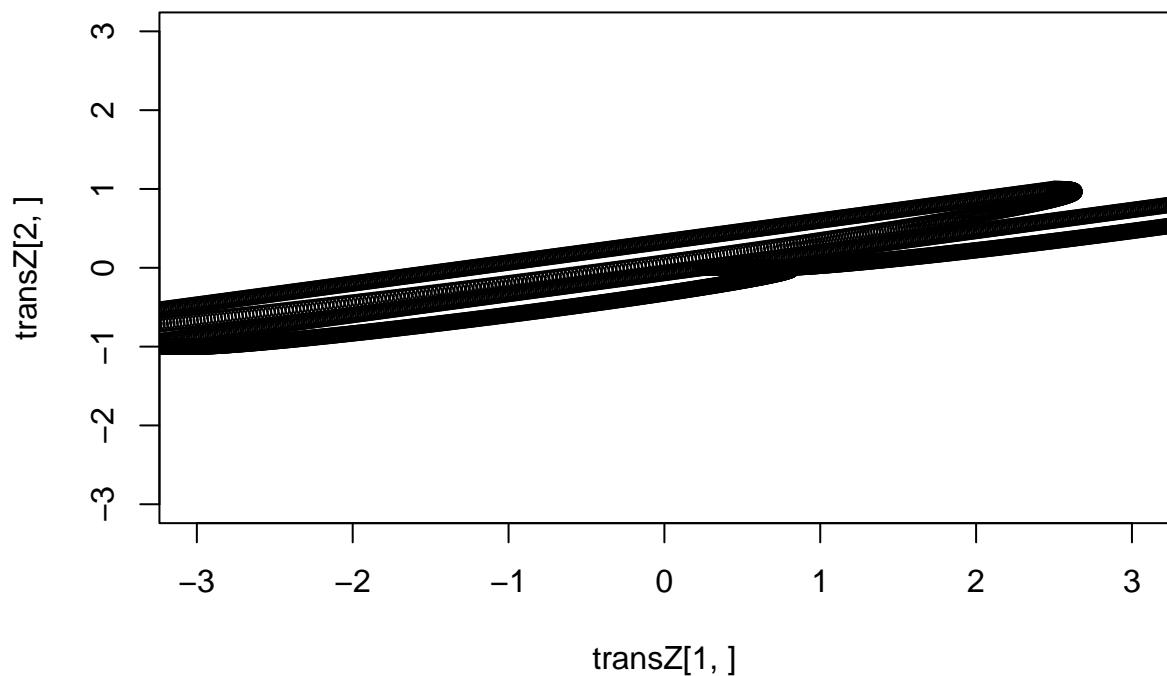


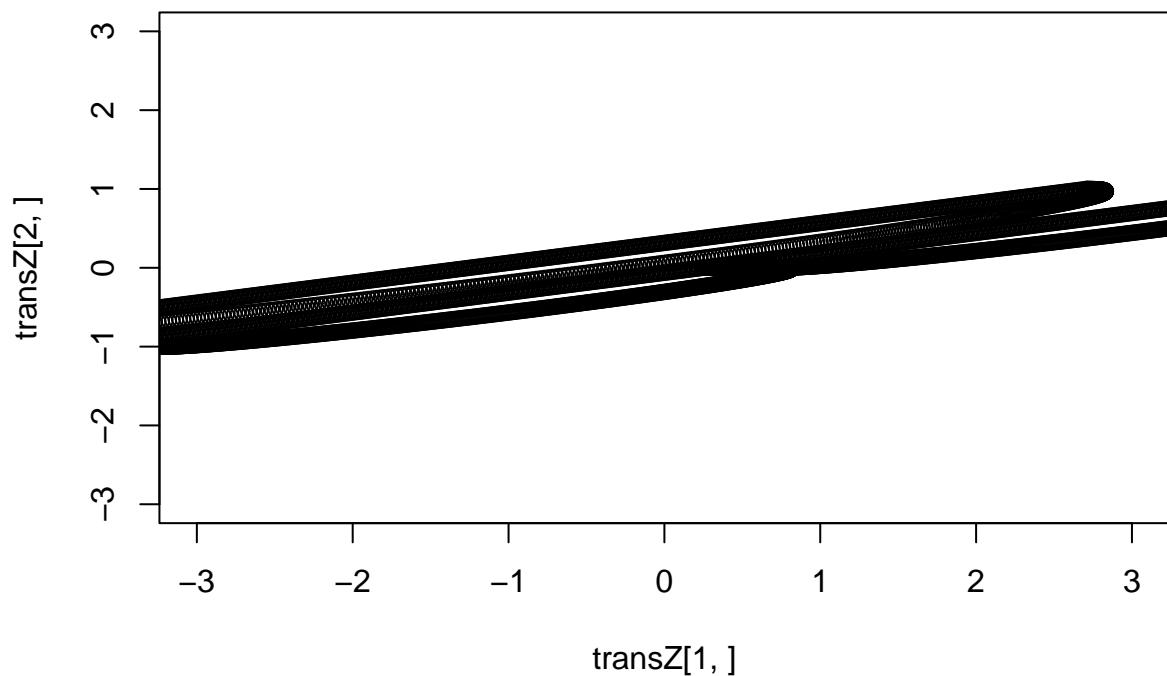


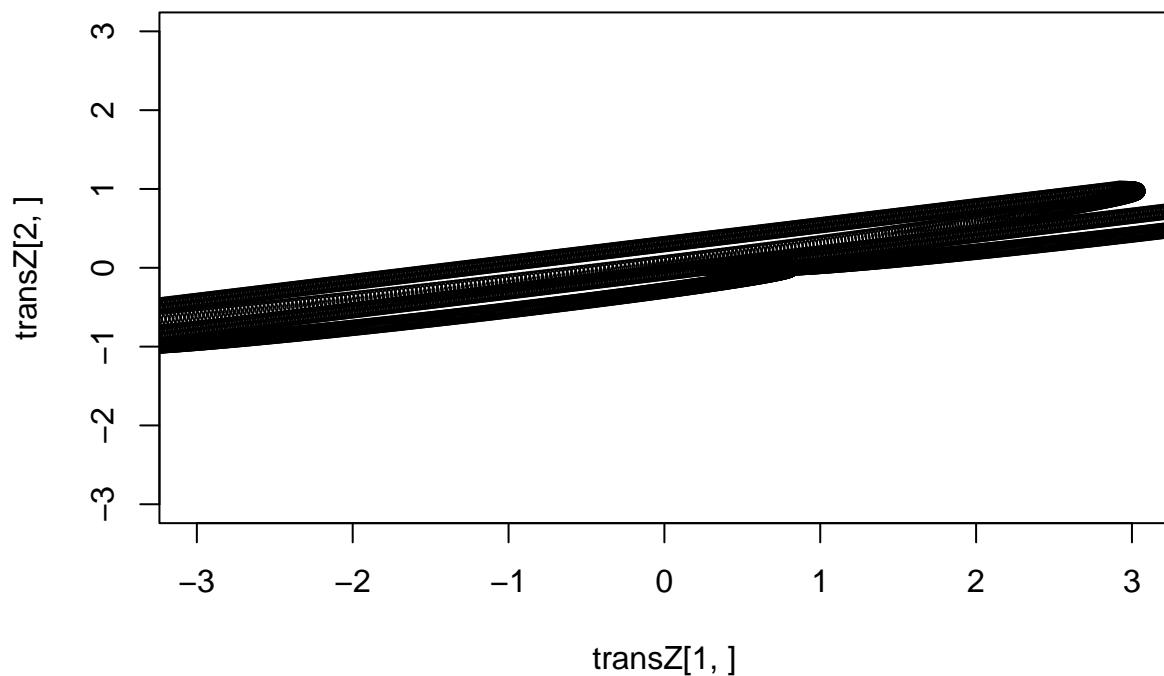


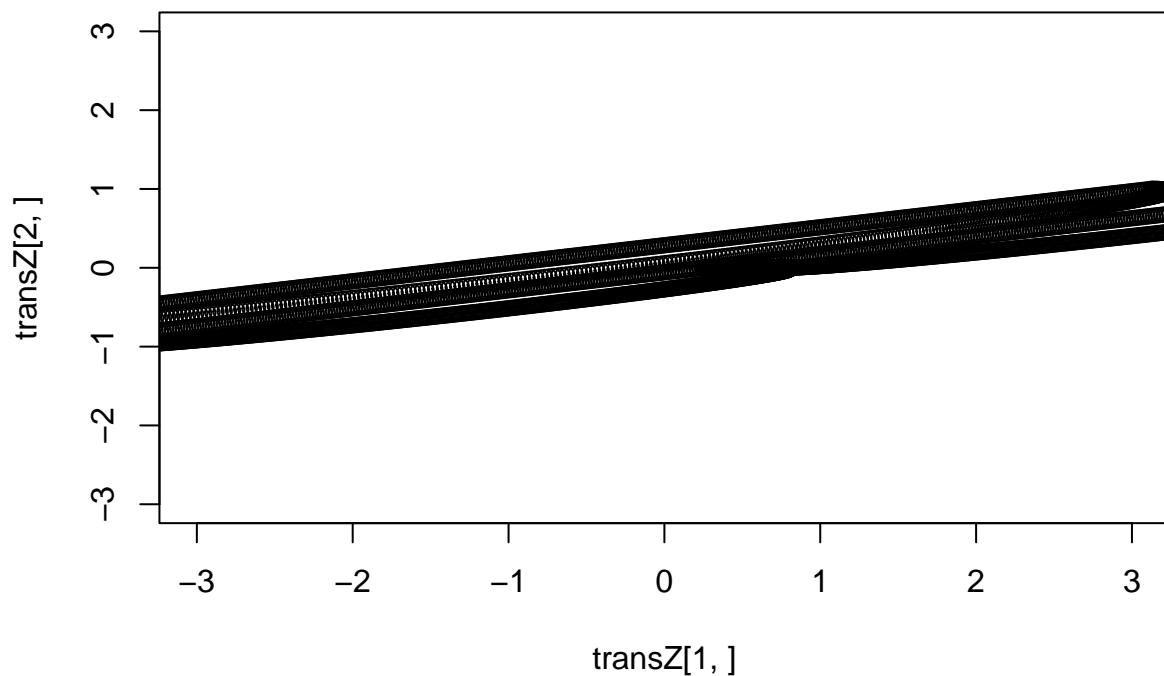


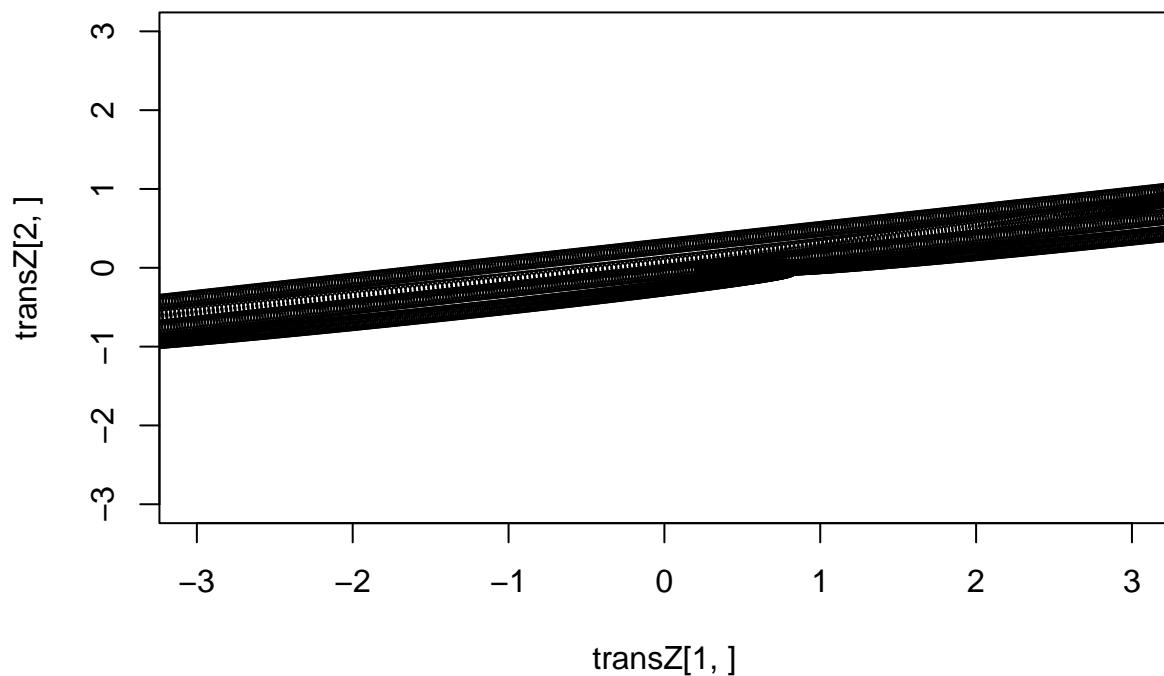


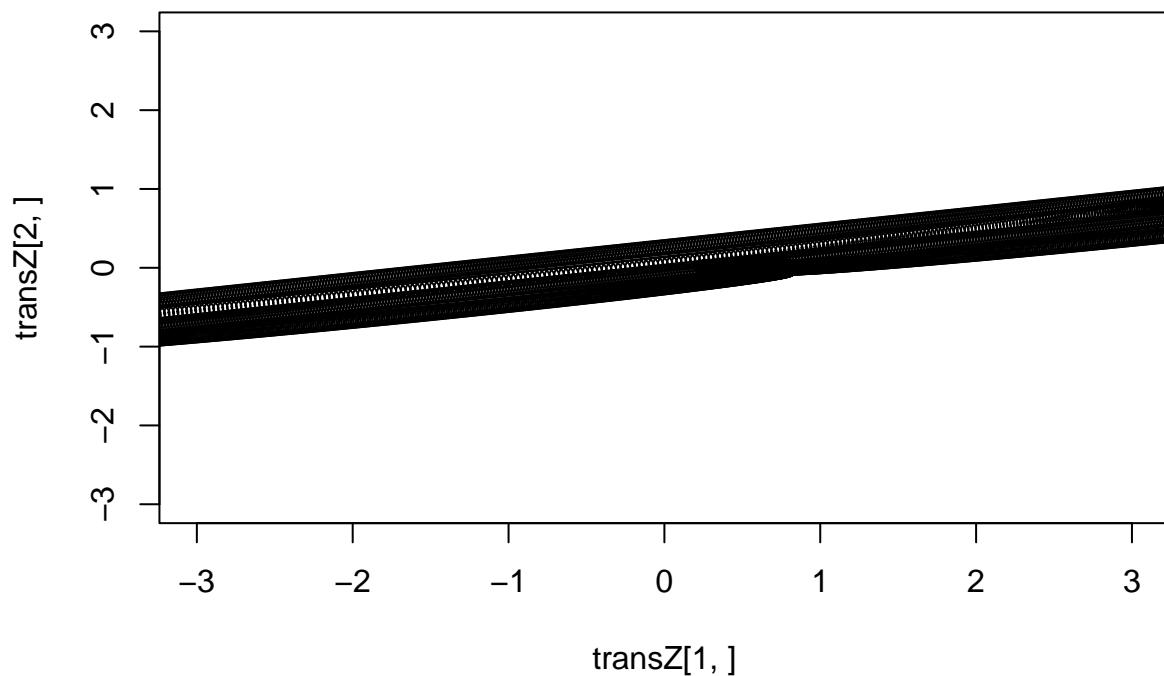


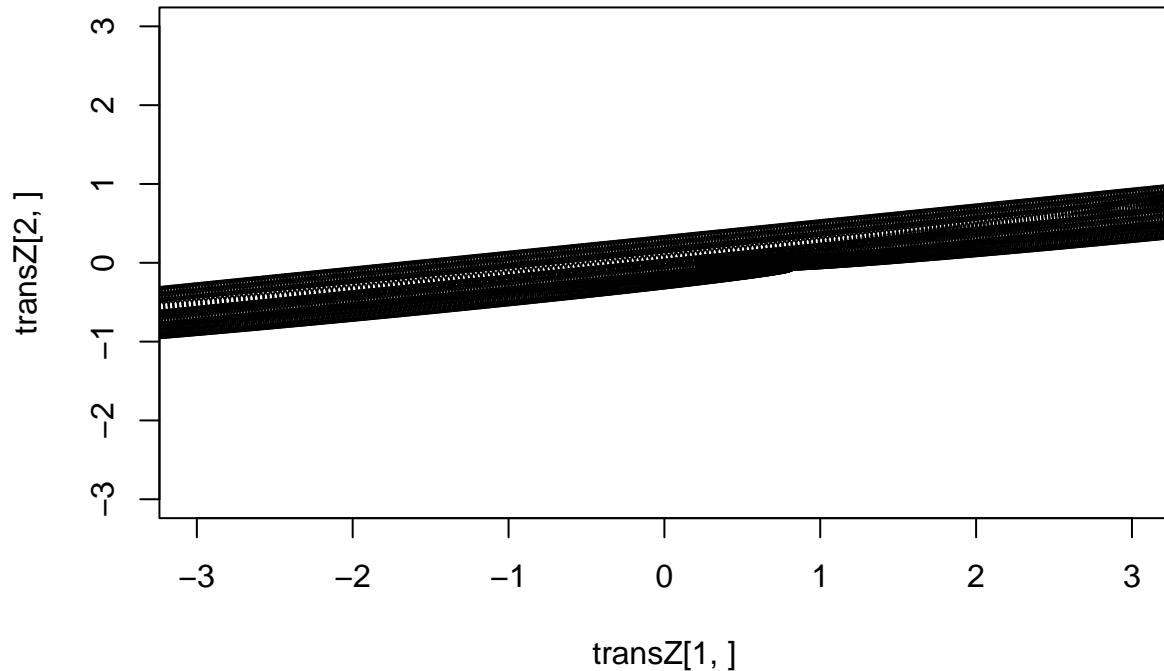










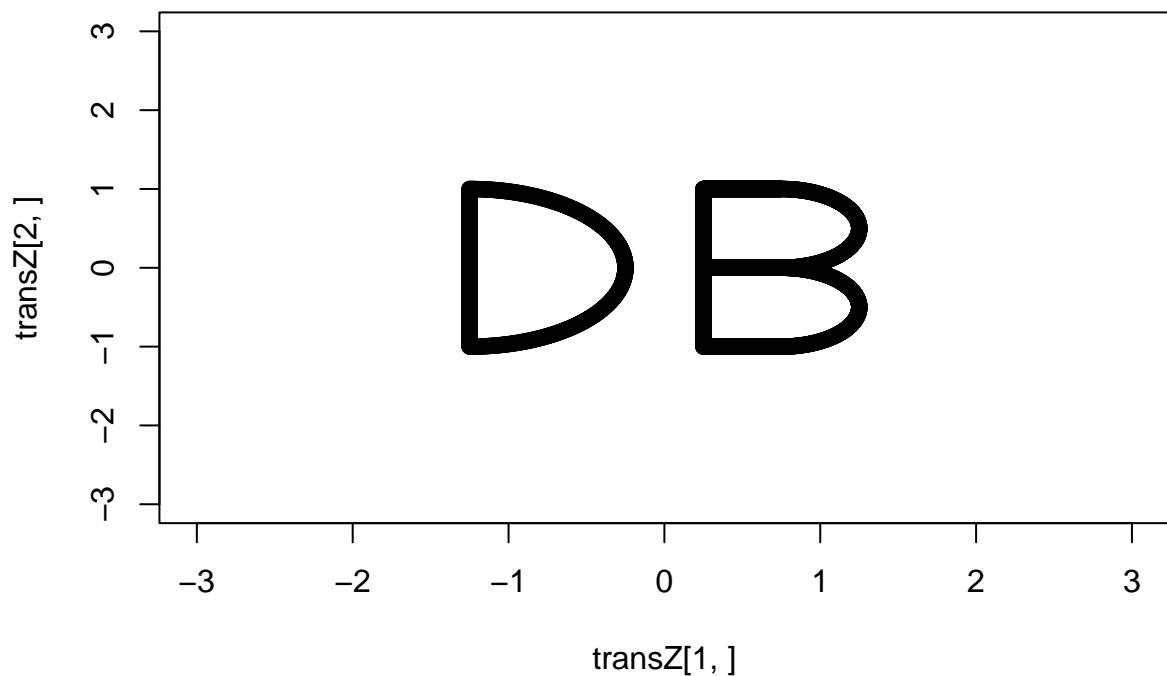


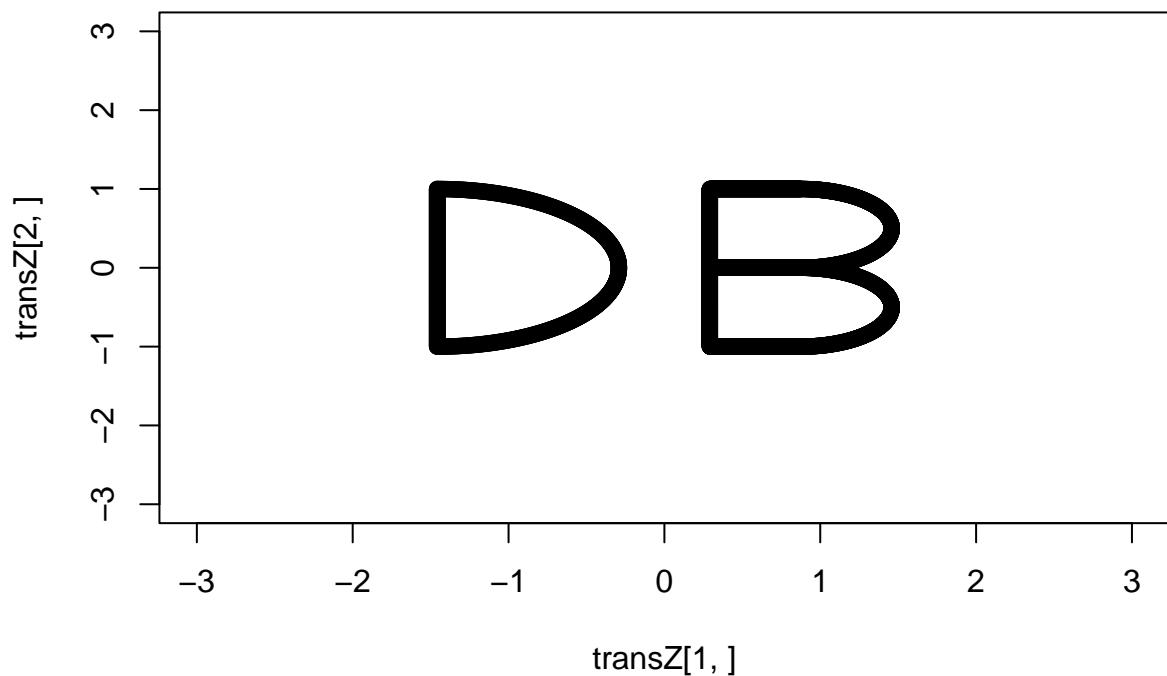
## Scaling

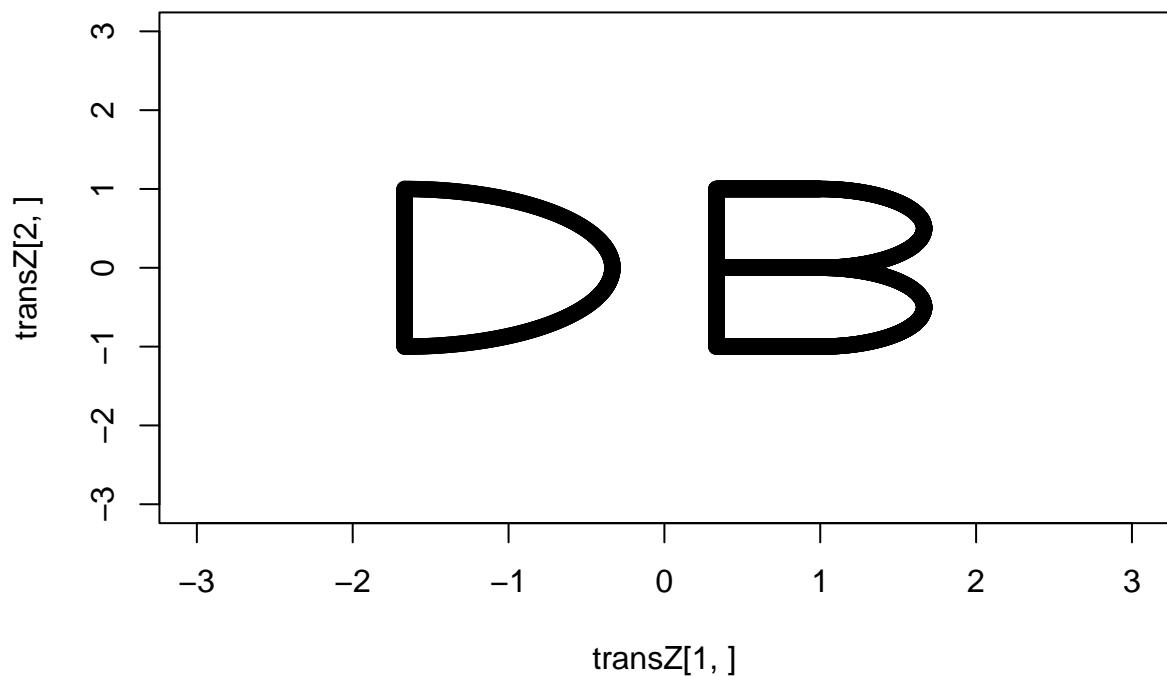
Scaling allows us to apply a scaling factor to x or y coordinate. If the scaling factor is less than 1, the image will shrink. If the scaling factor is greater than 1, the image will stretch.

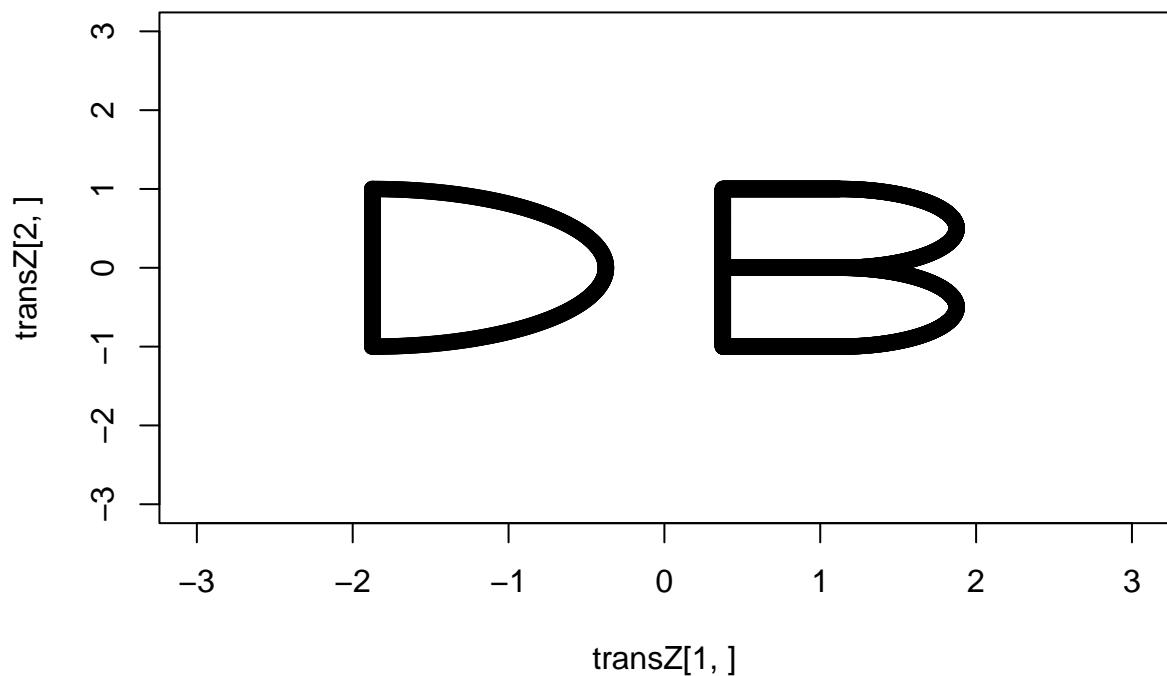
### Stretching horizontally

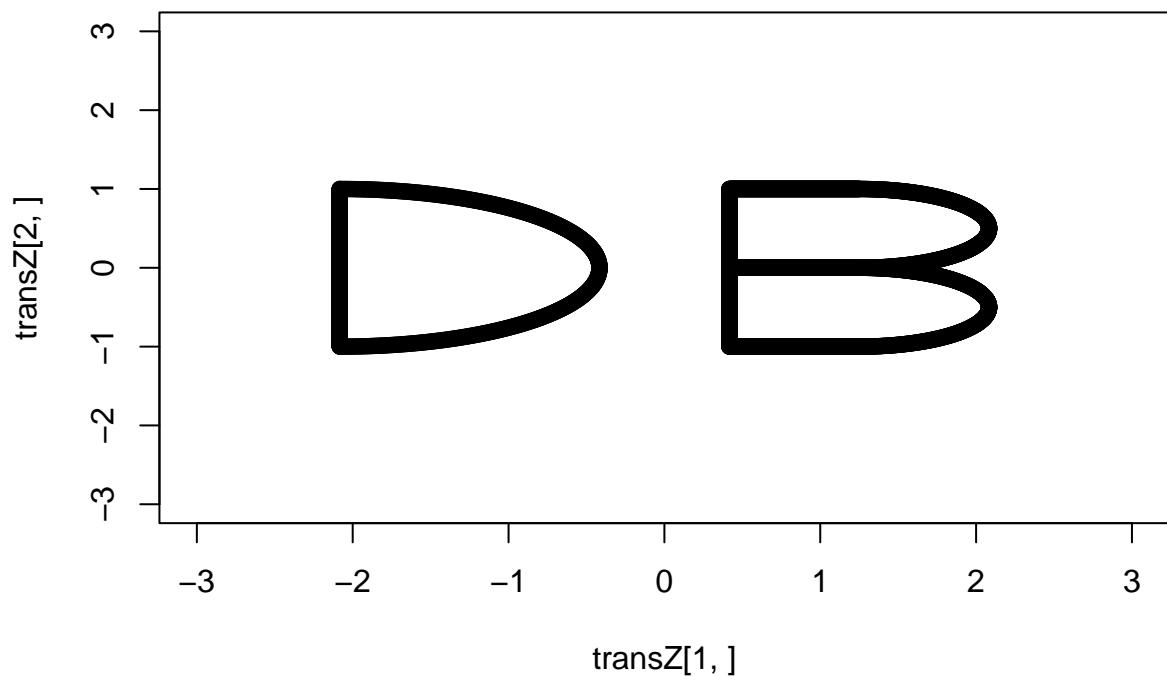
```
x11()
for (scaling_factor in seq(1,5,length.out=25)) {
  transZ <- apply(z,2,function(x) leftMultiply(x,matrix(c(scaling_factor,0,0,1),nrow = 2, ncol = 2)))
  plot(transZ[2,]~transZ[1,], xlim = c(-3,3), ylim = c(-3,3))
}
```

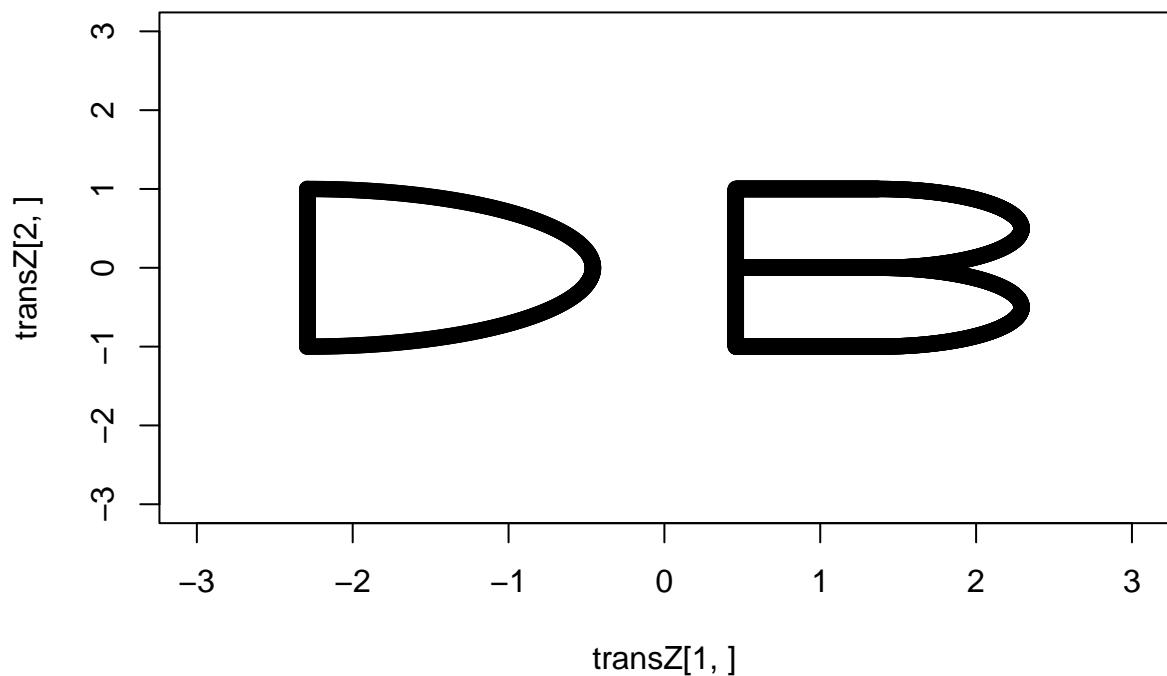


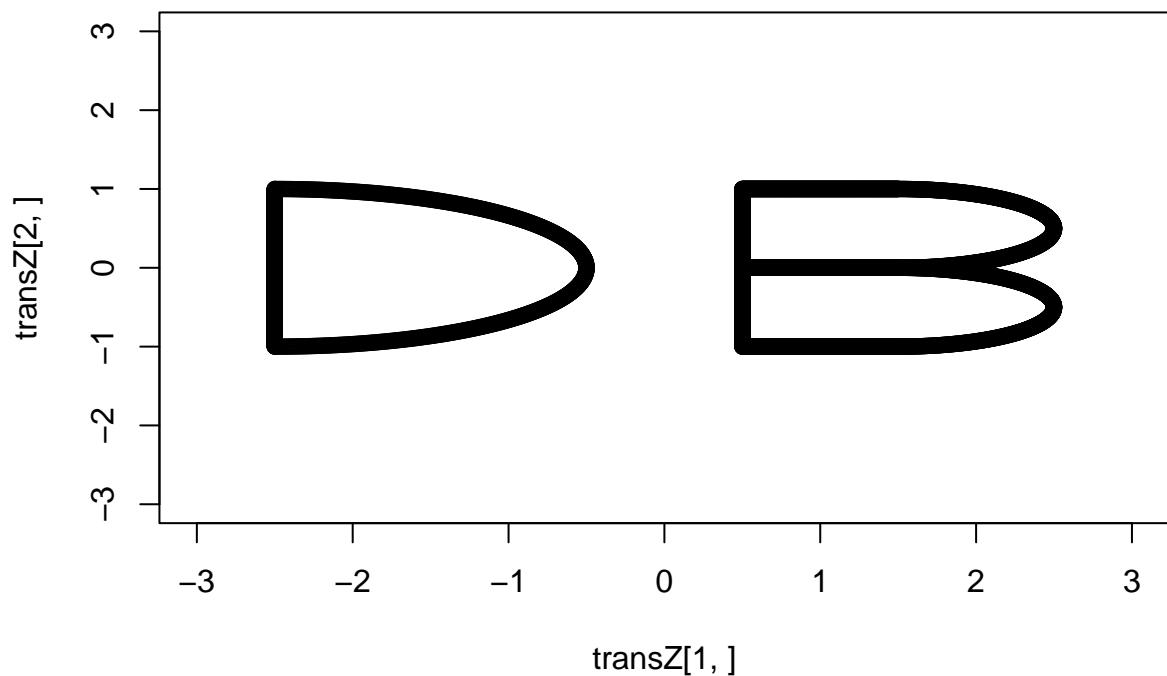


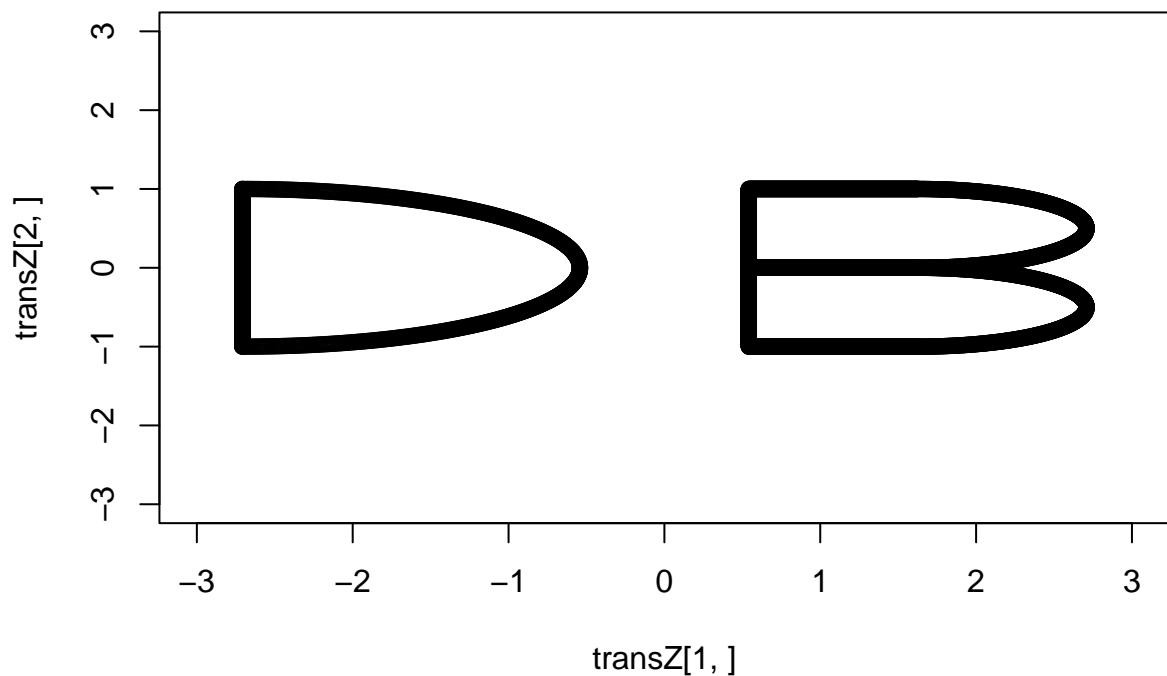


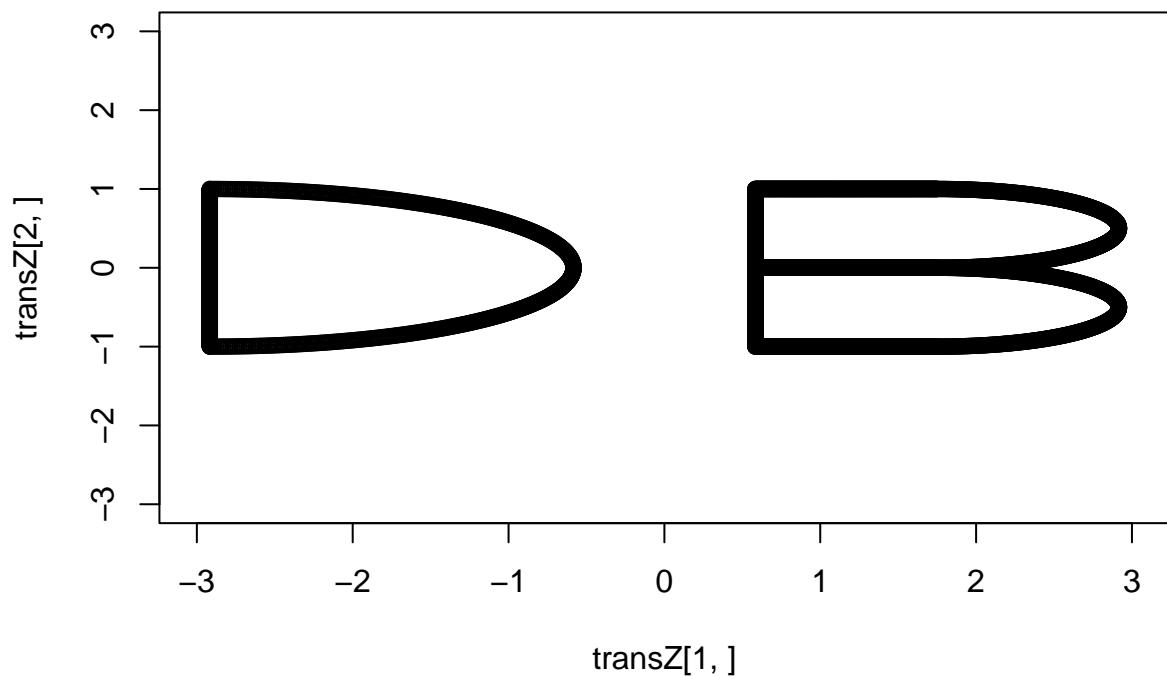


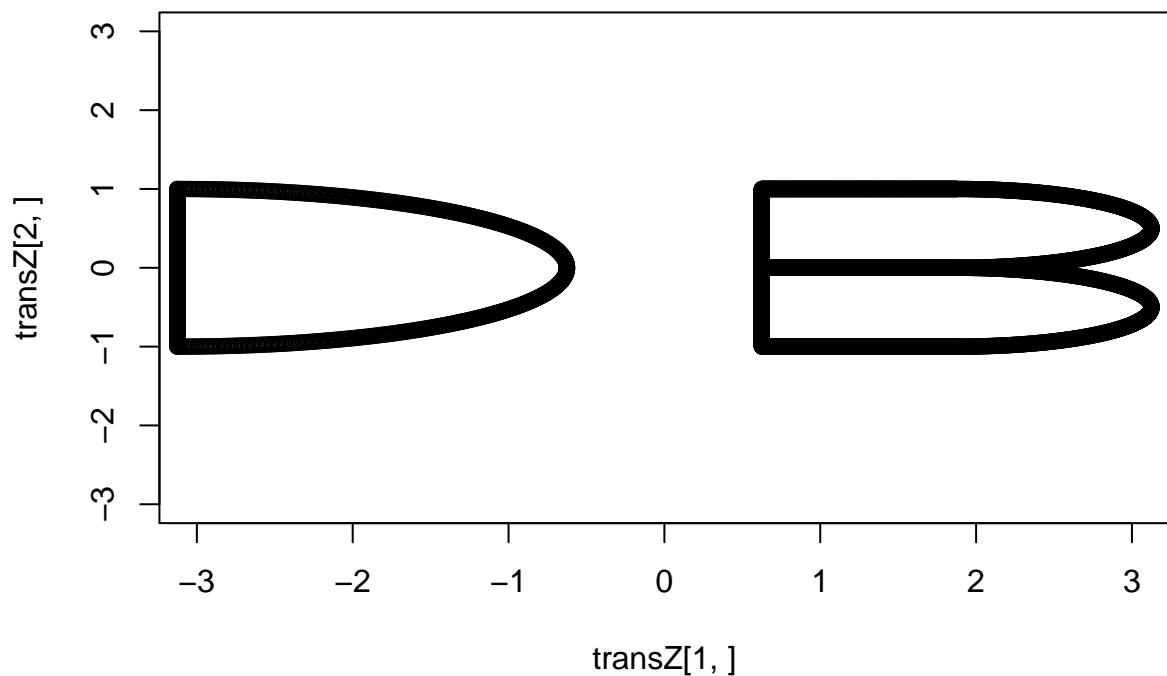


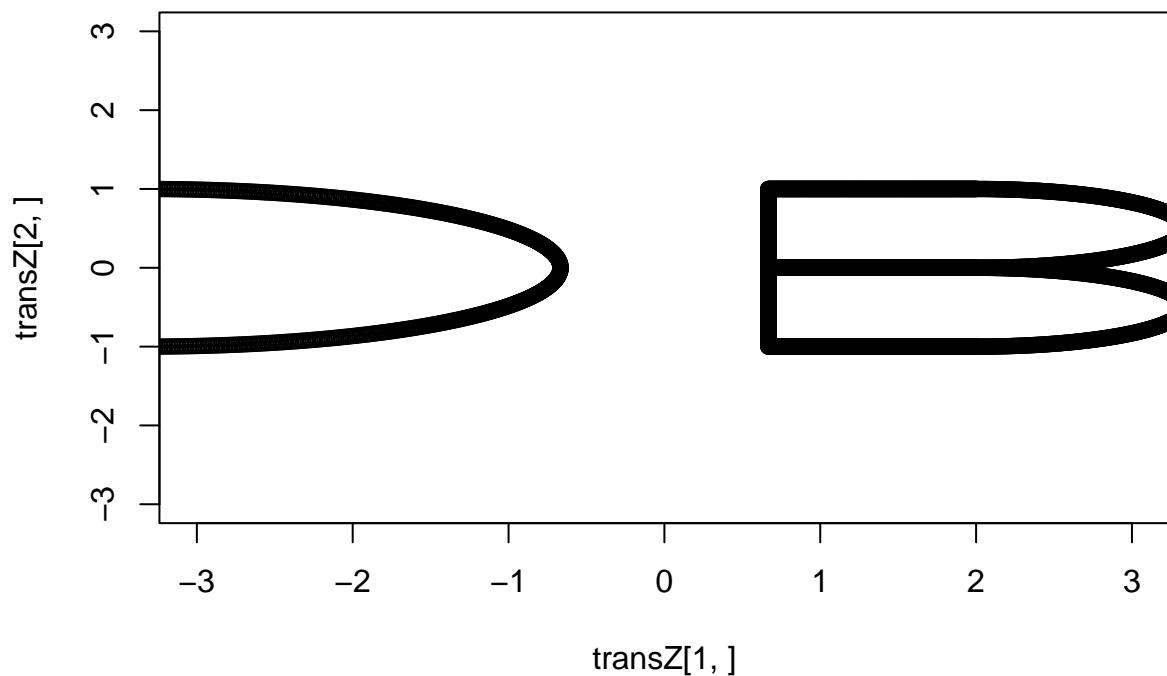


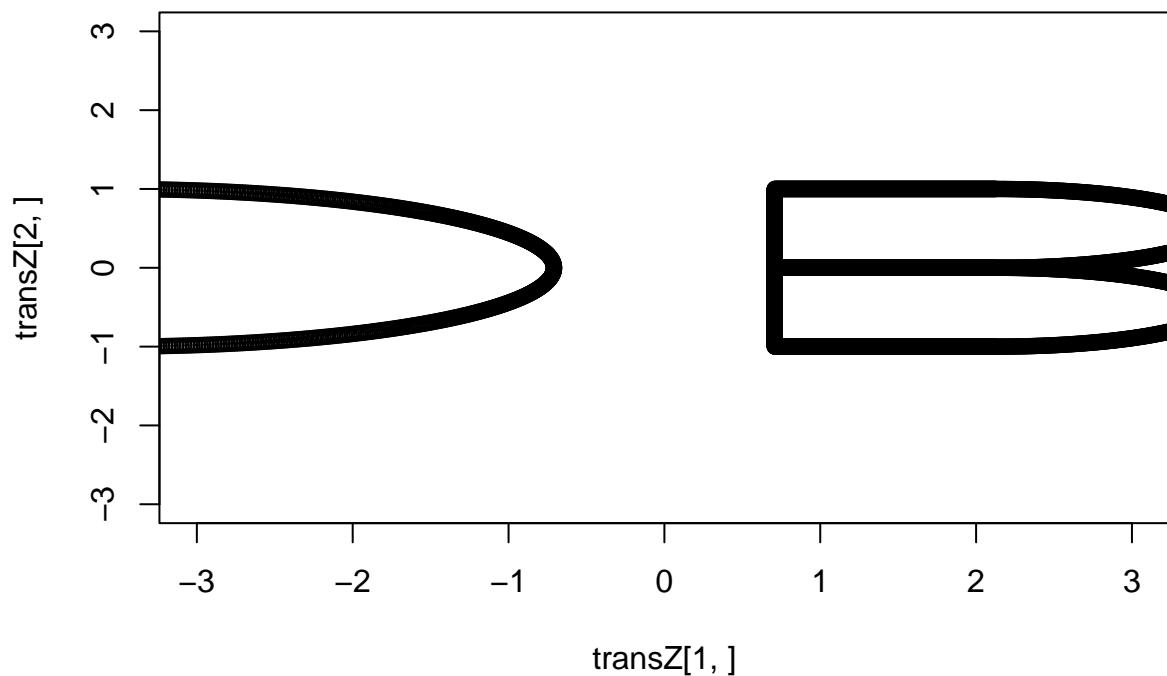


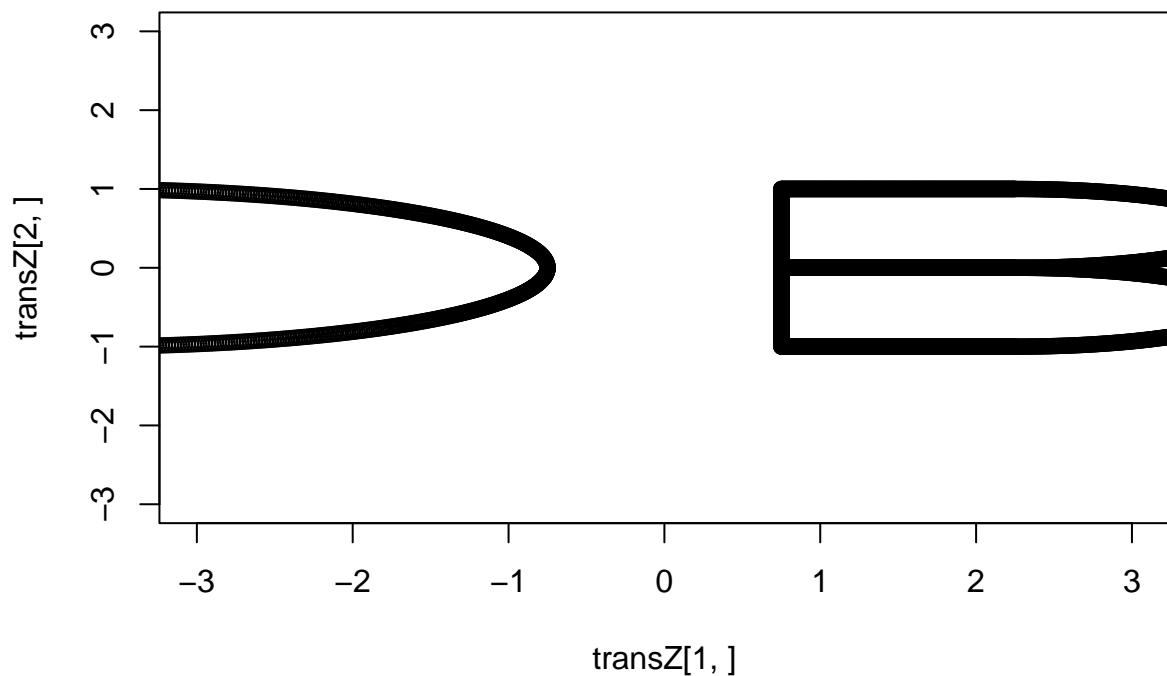


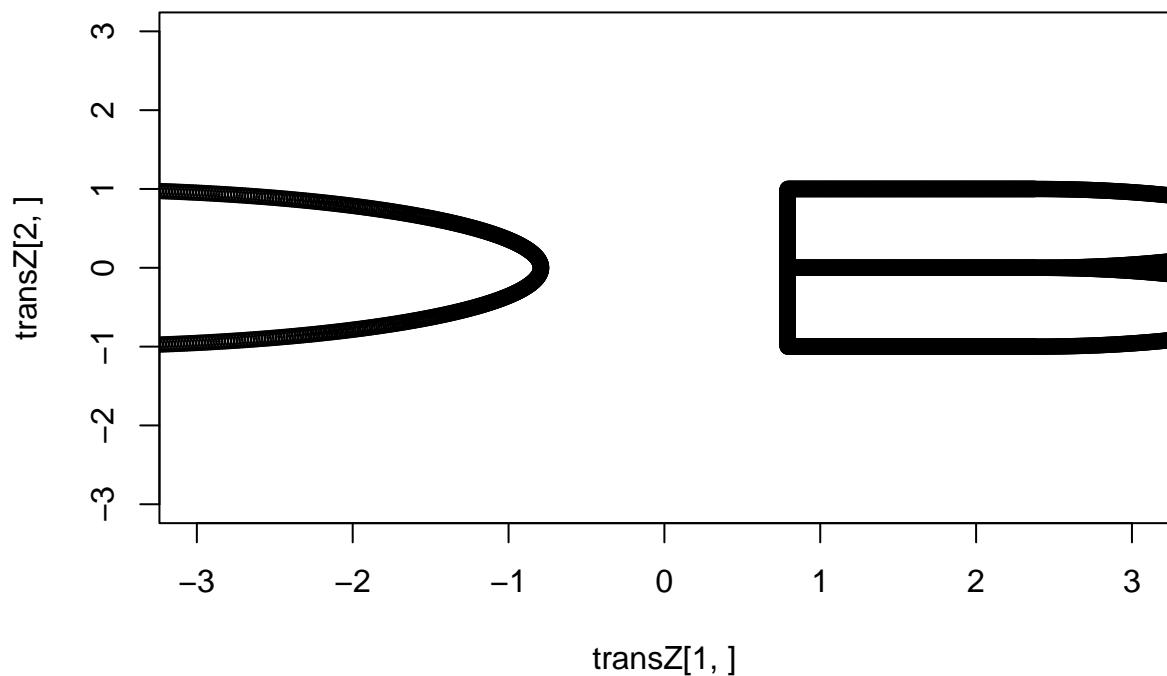


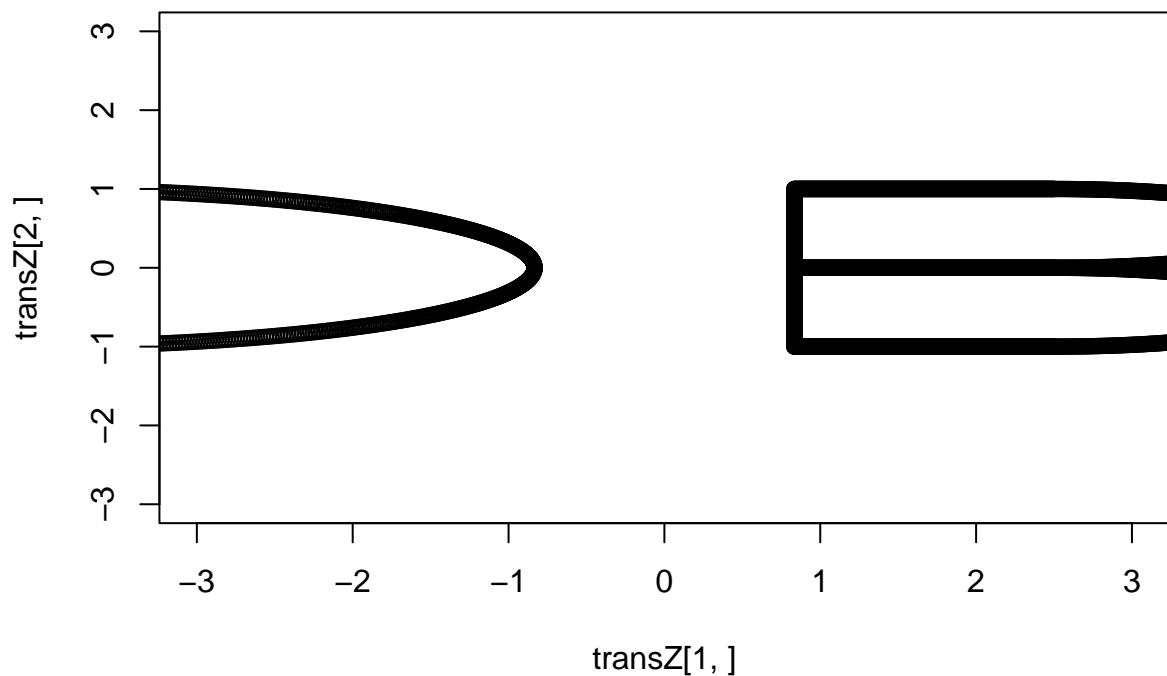


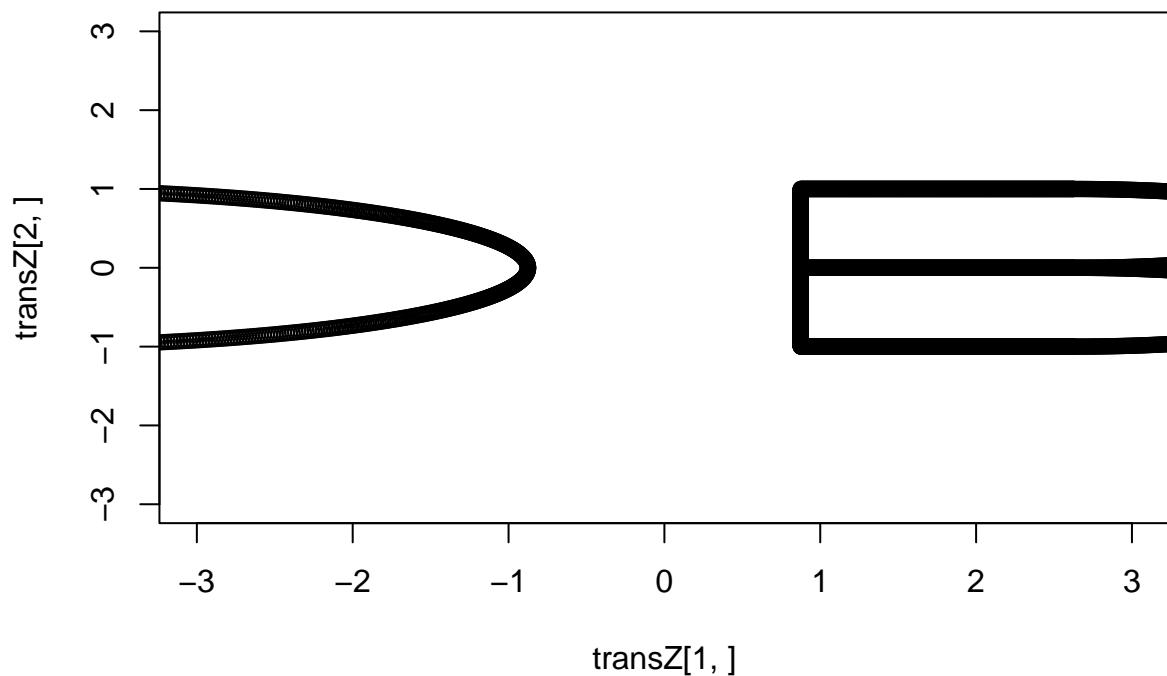


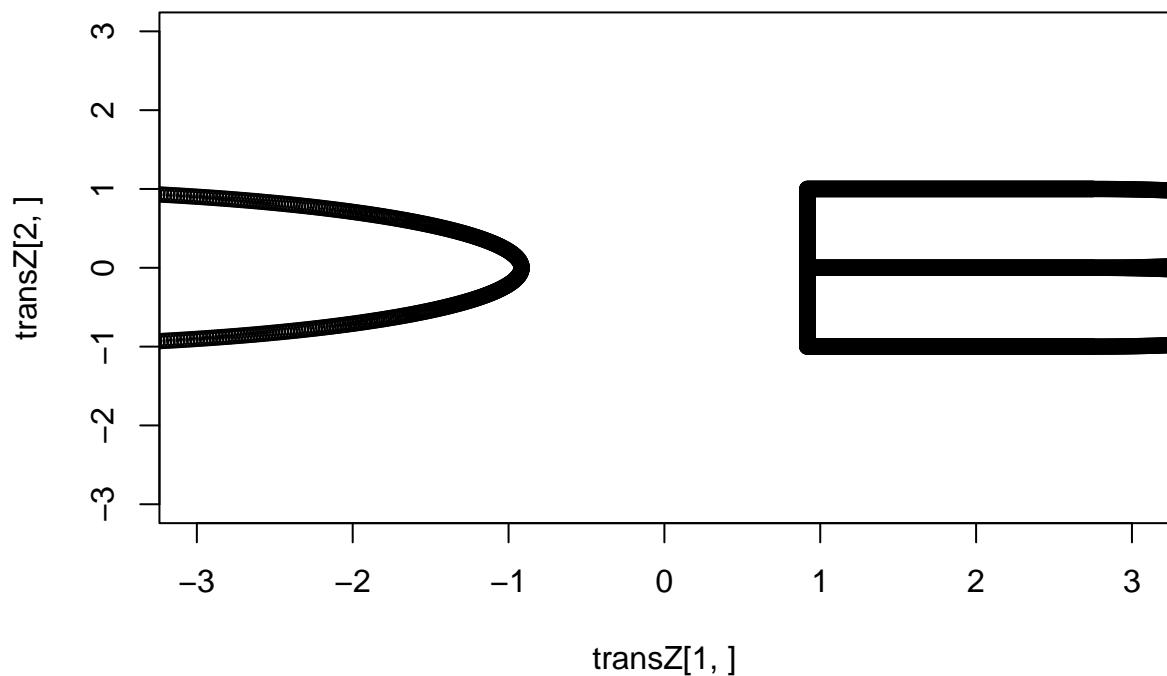


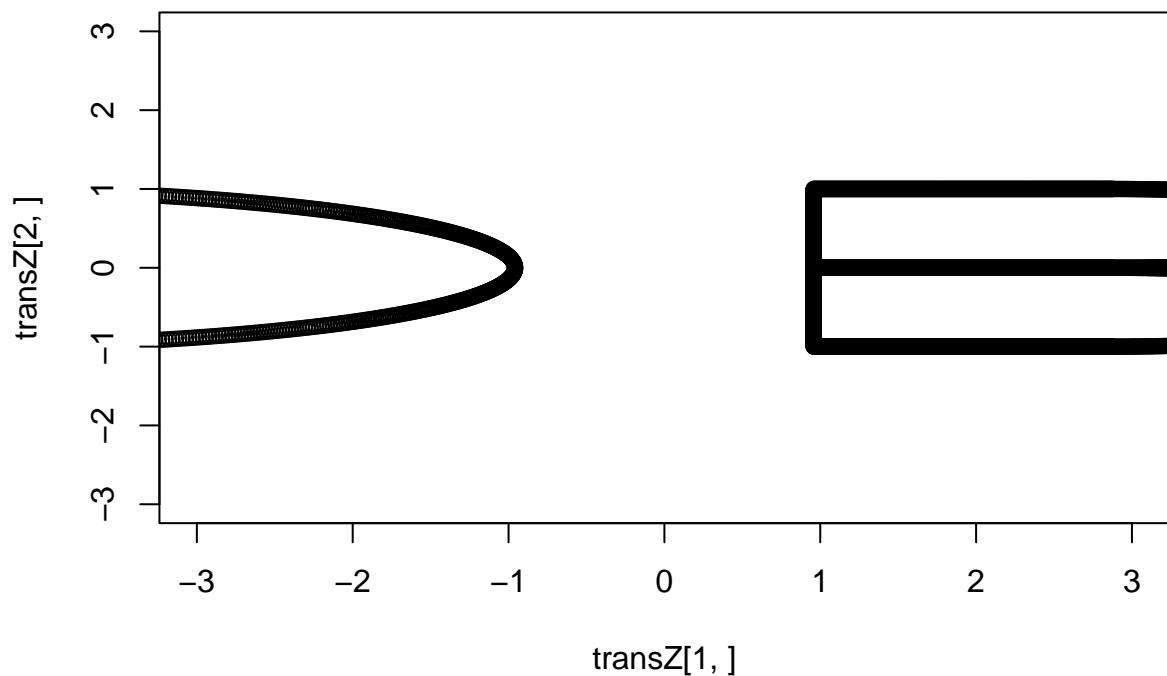


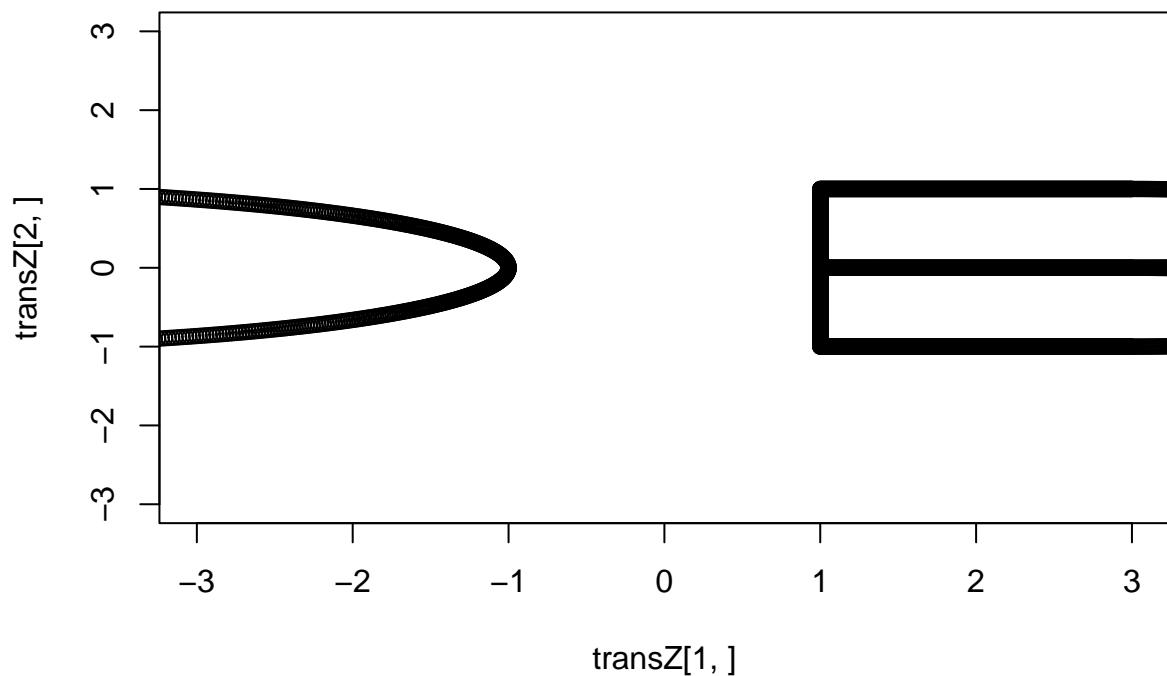


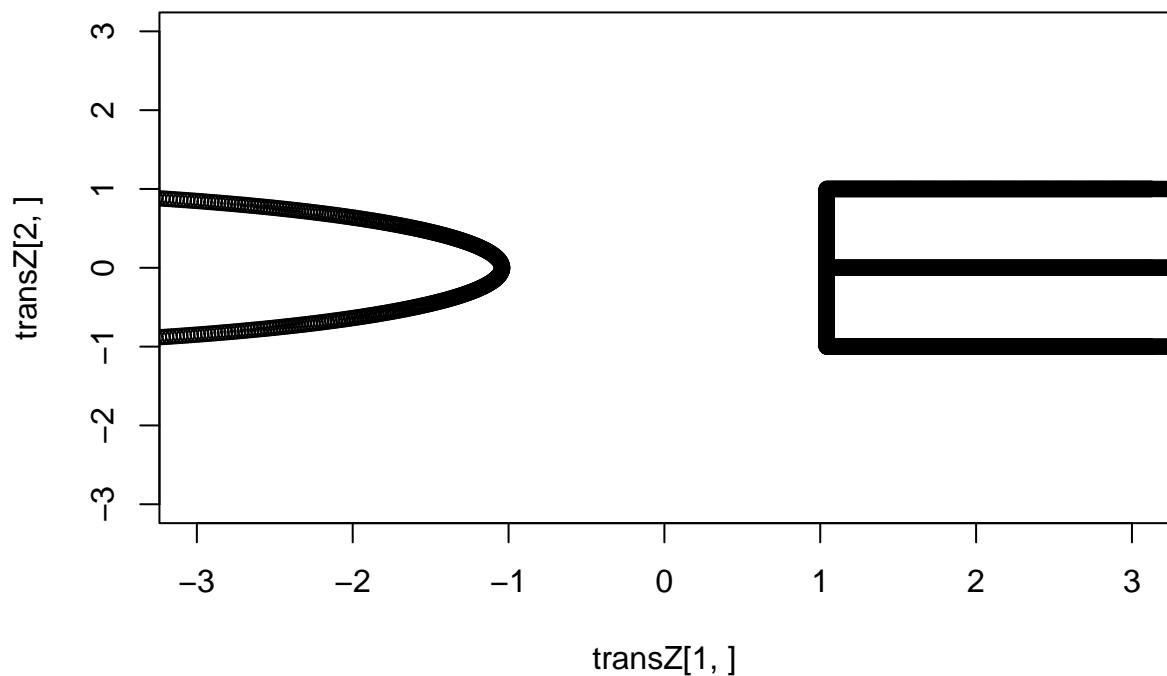


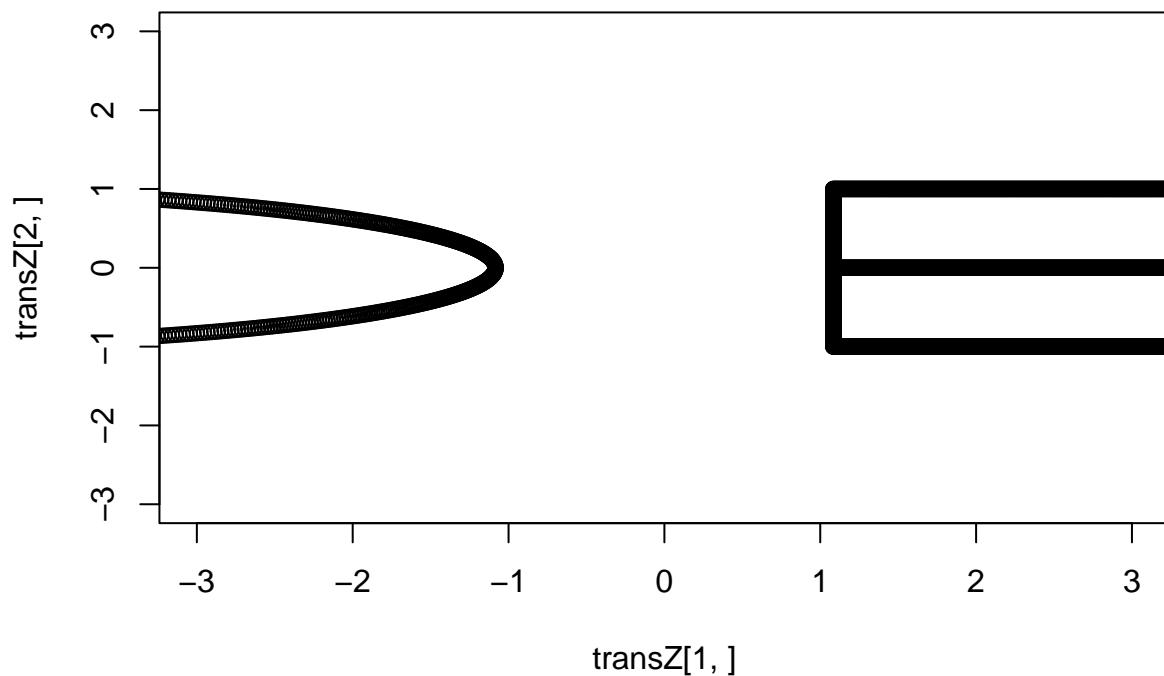


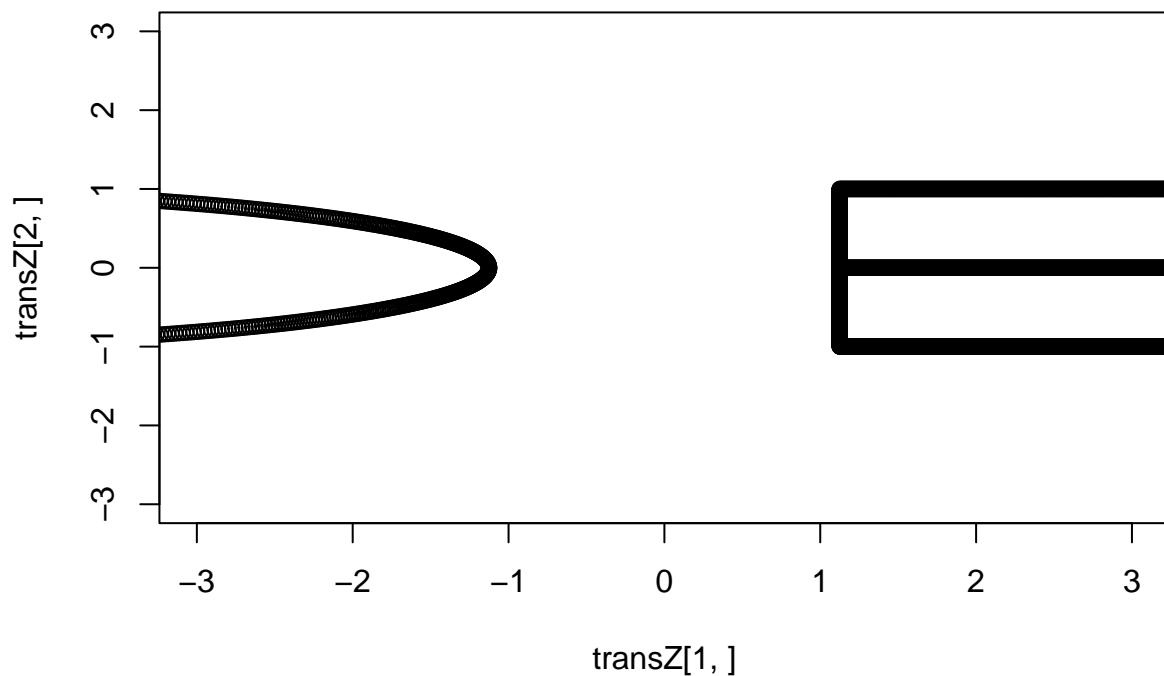


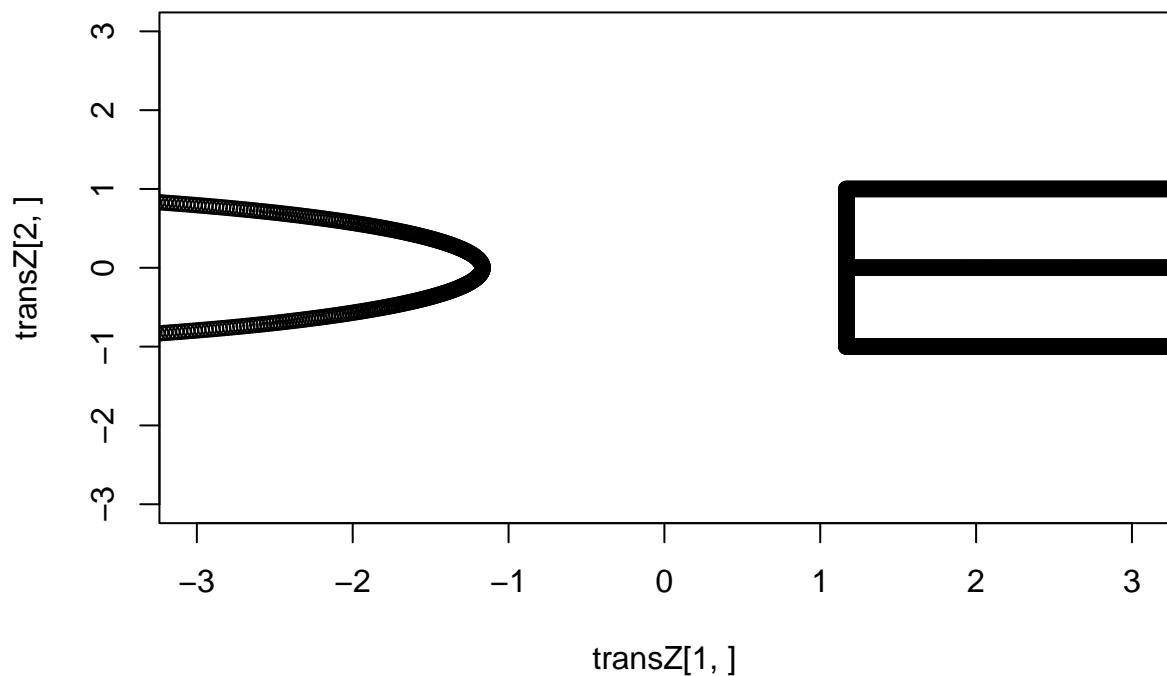


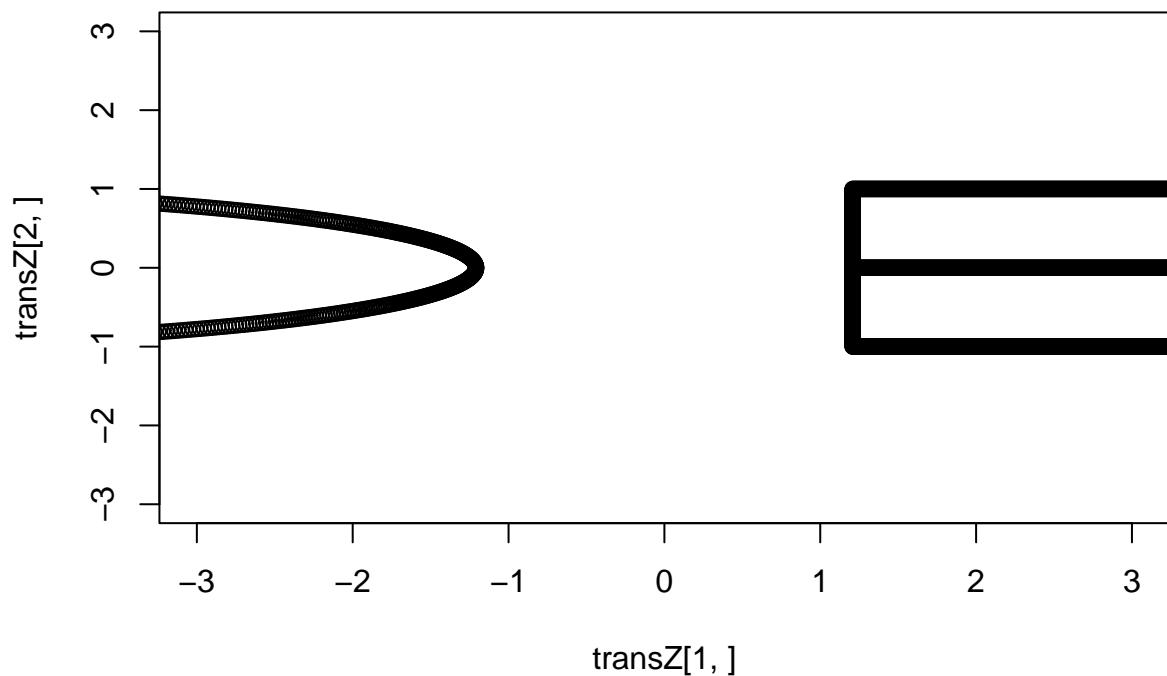


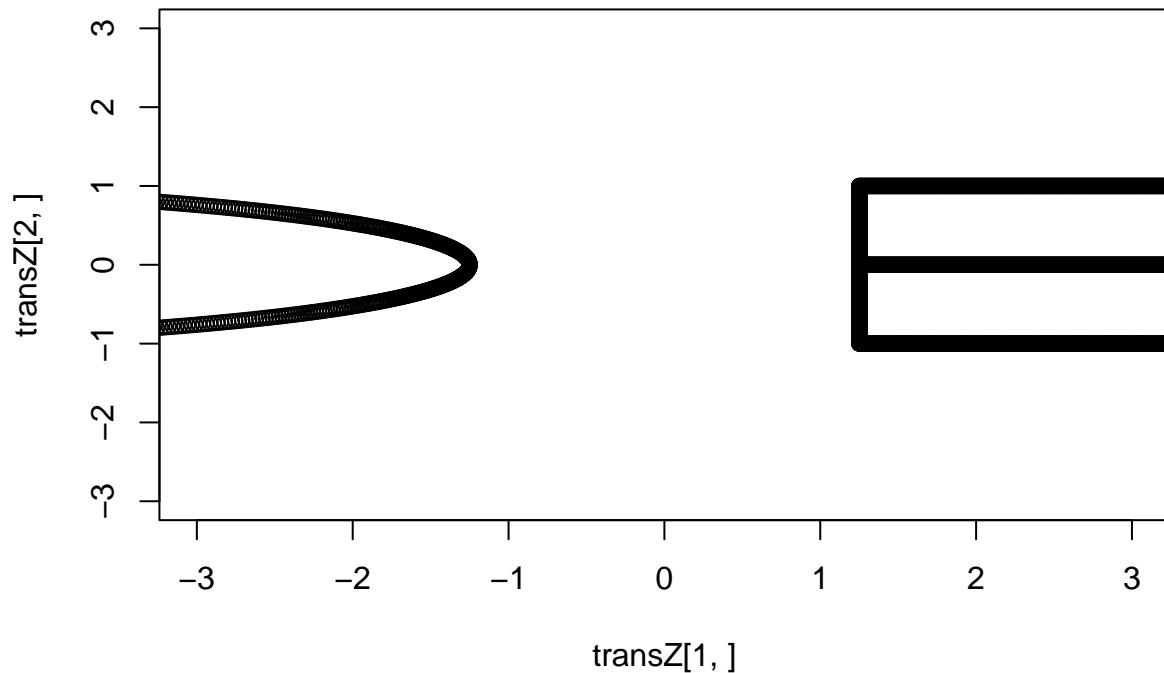






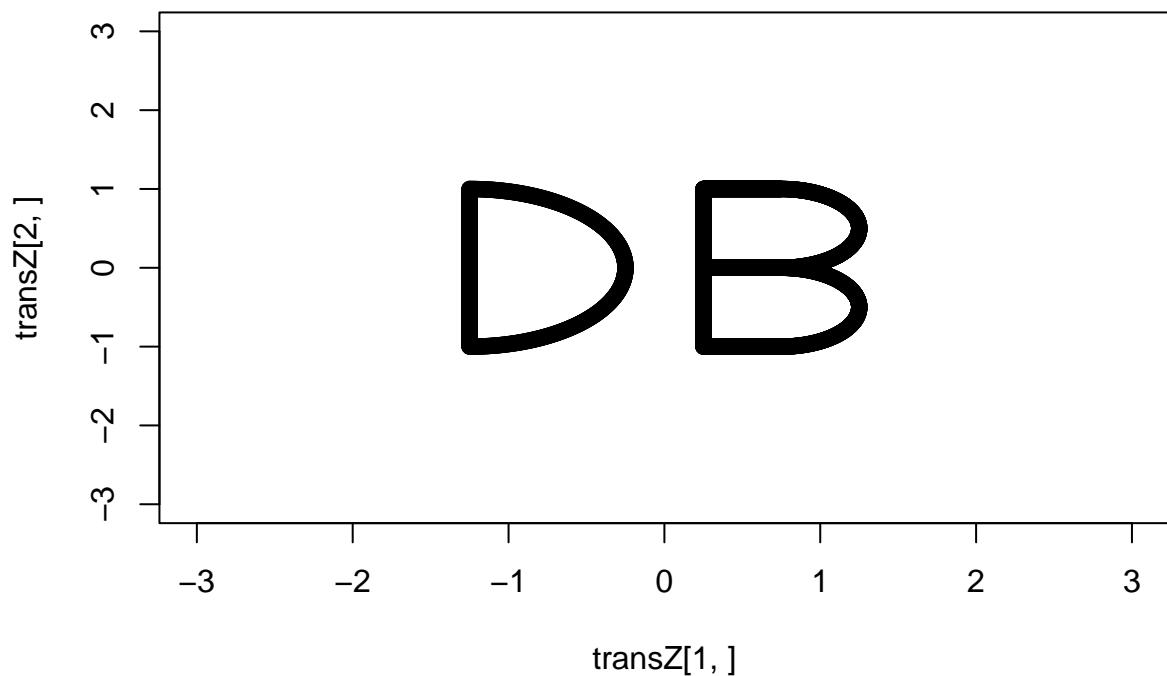


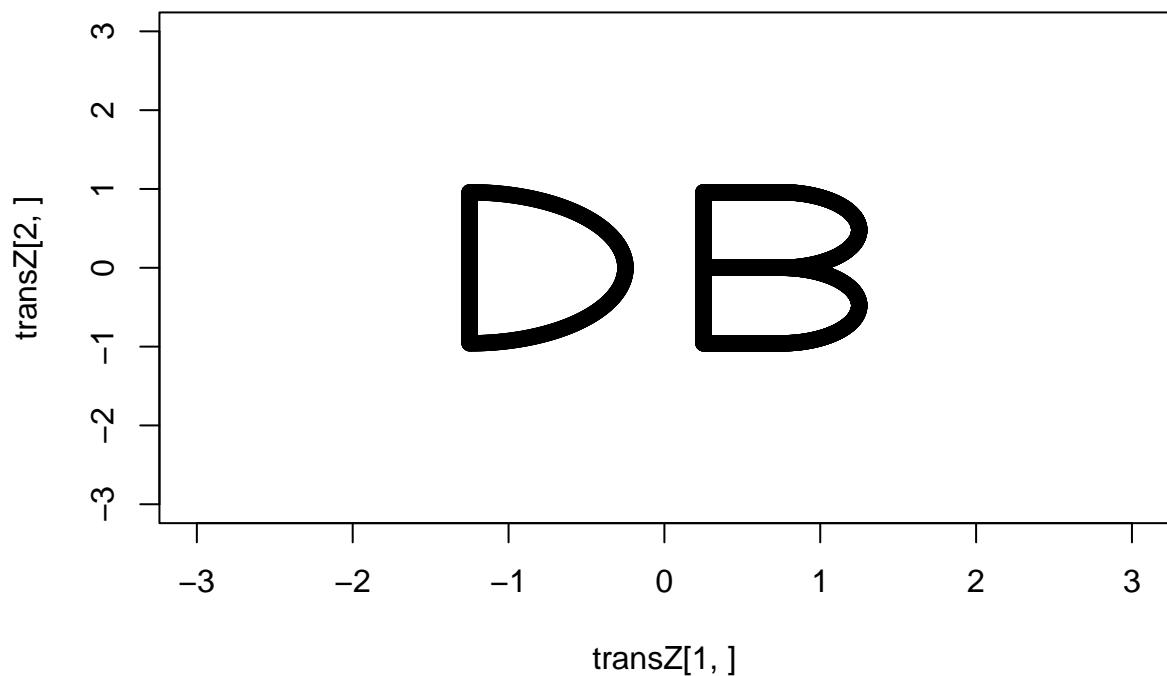


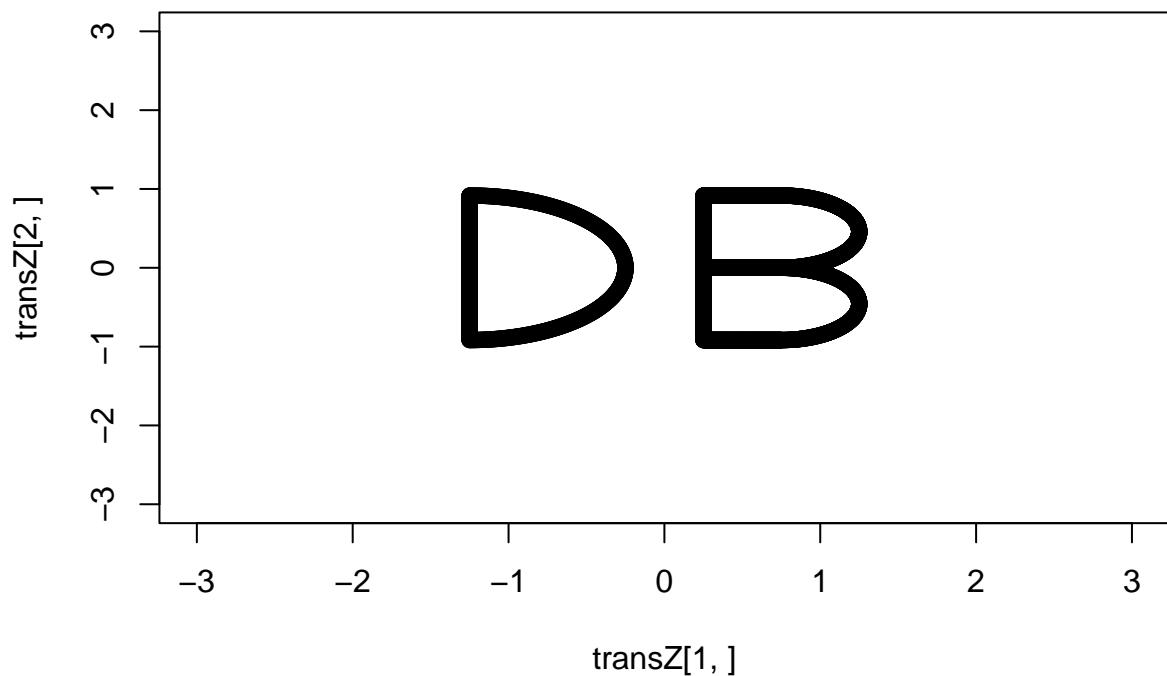


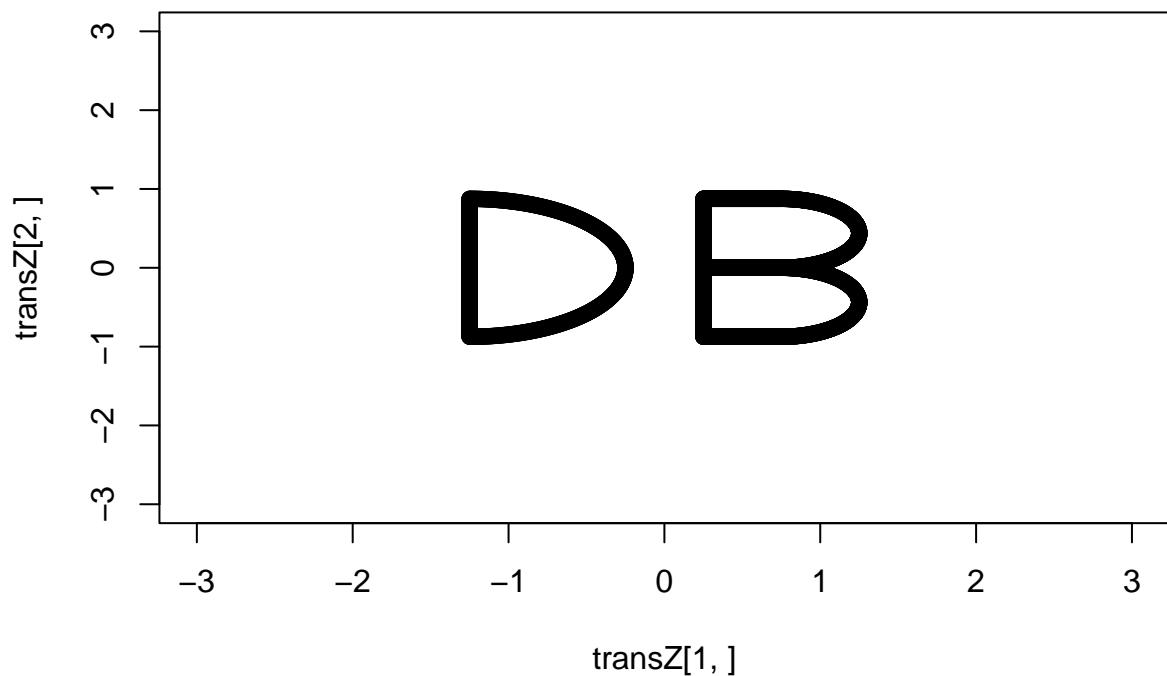
Shrinking vertically

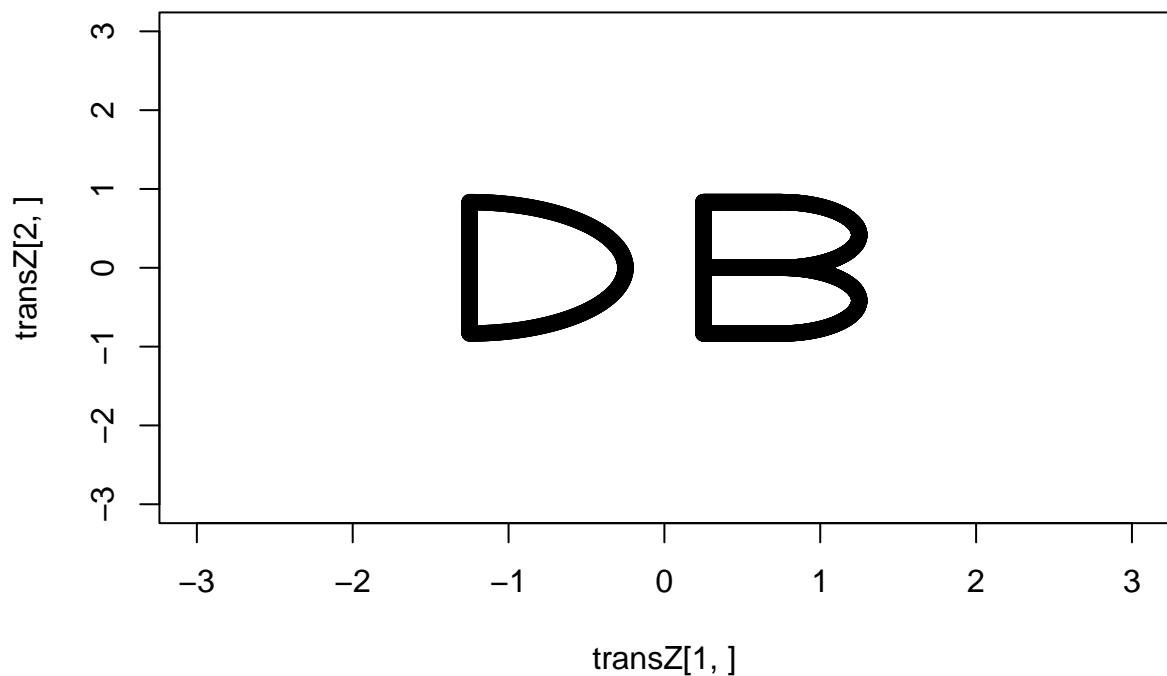
```
x11()
for (scaling_factor in seq(1,0,length.out=25)) {
  transZ <- apply(z,2,function(x) leftMultiply(x,matrix(c(1,0,0,scaling_factor),nrow = 2, ncol = 2)))
  plot(transZ[2,]~transZ[1,], xlim = c(-3,3), ylim = c(-3,3))
}
```

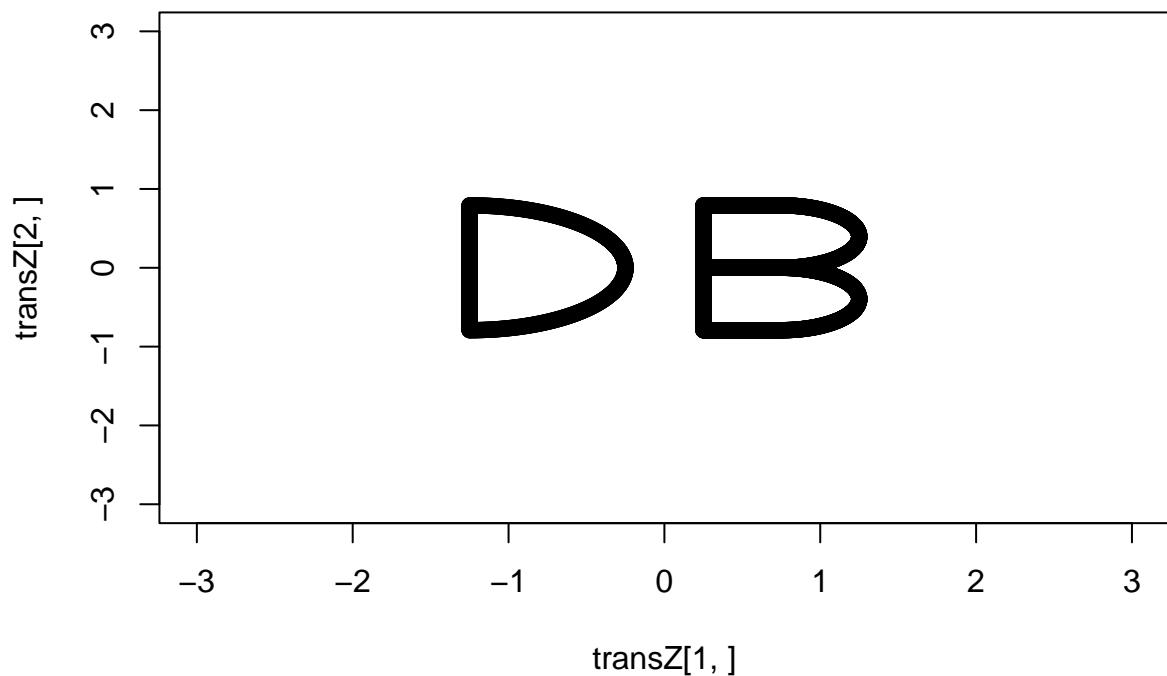


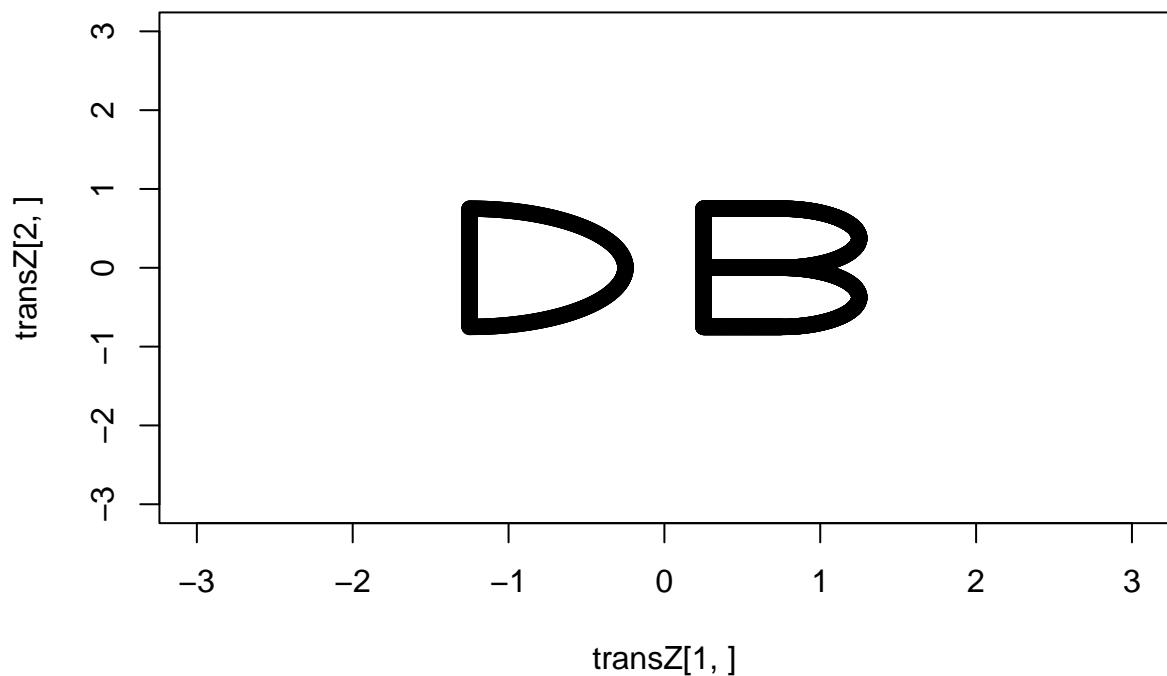


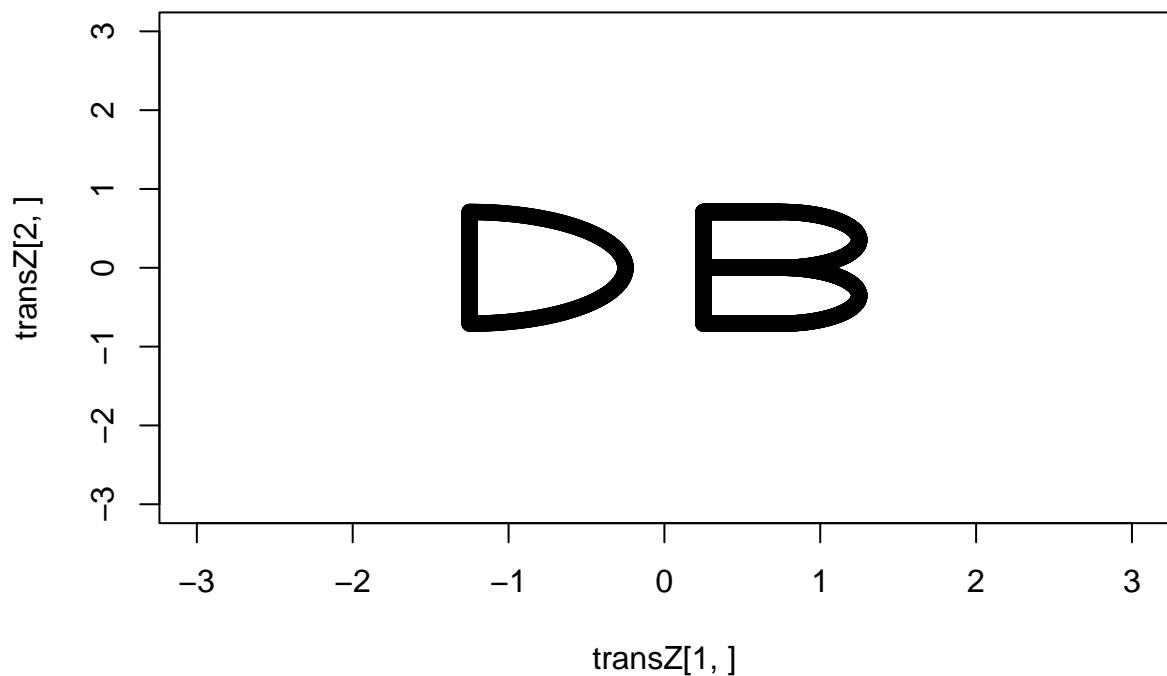


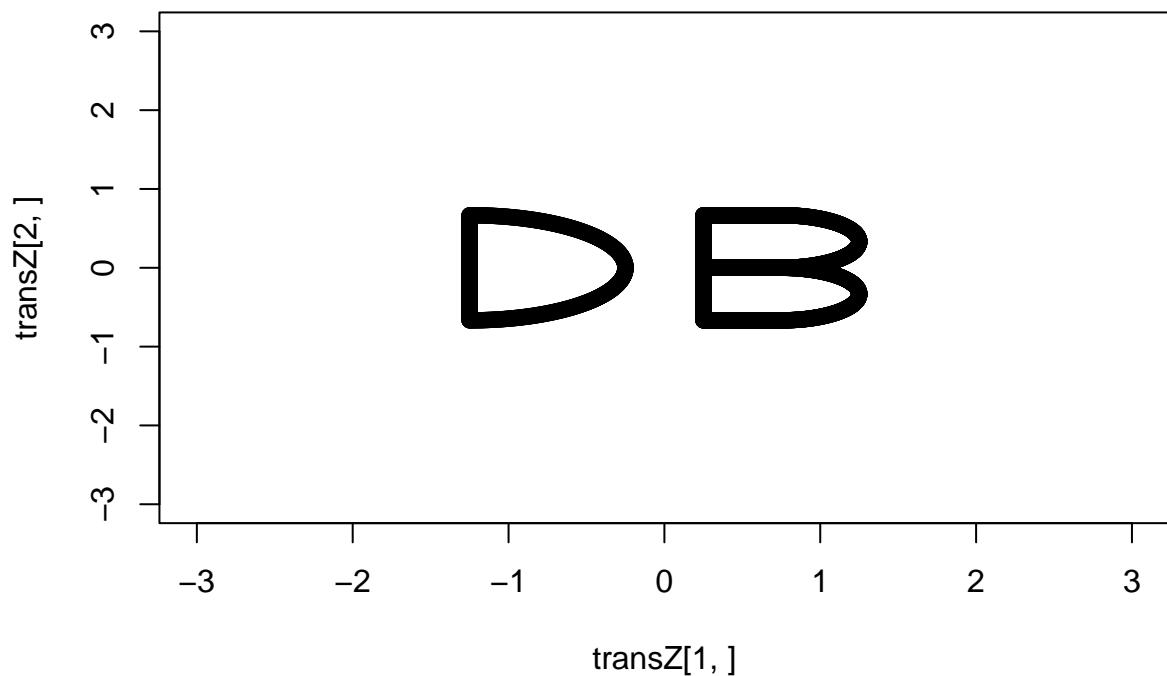


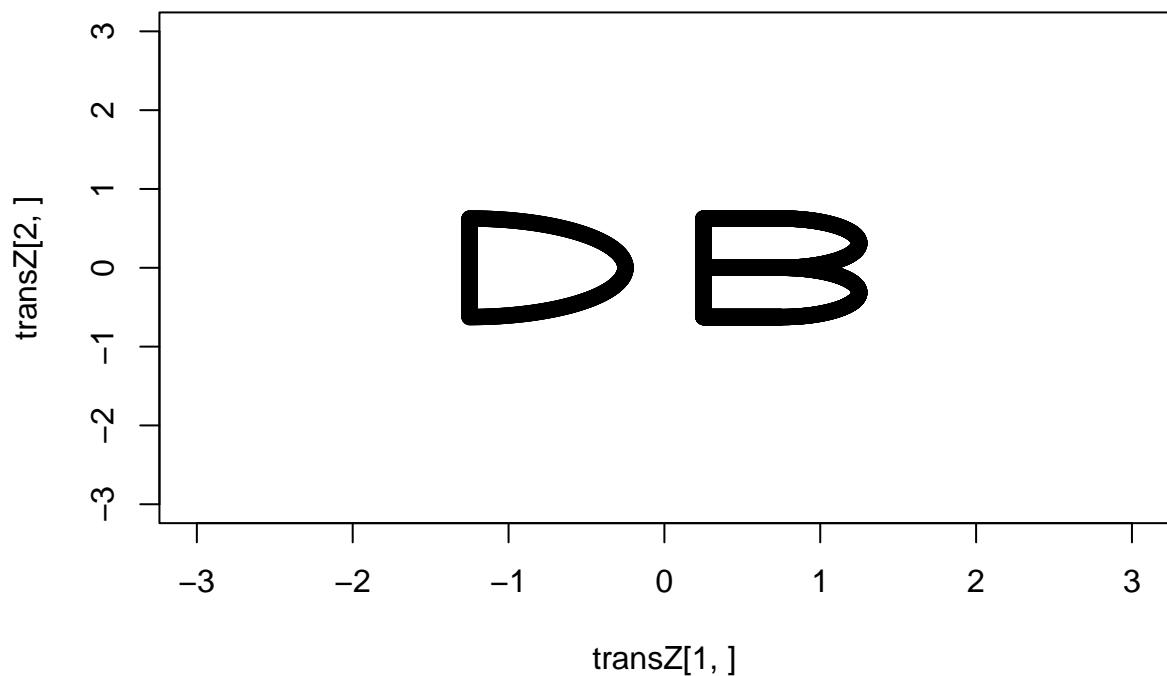


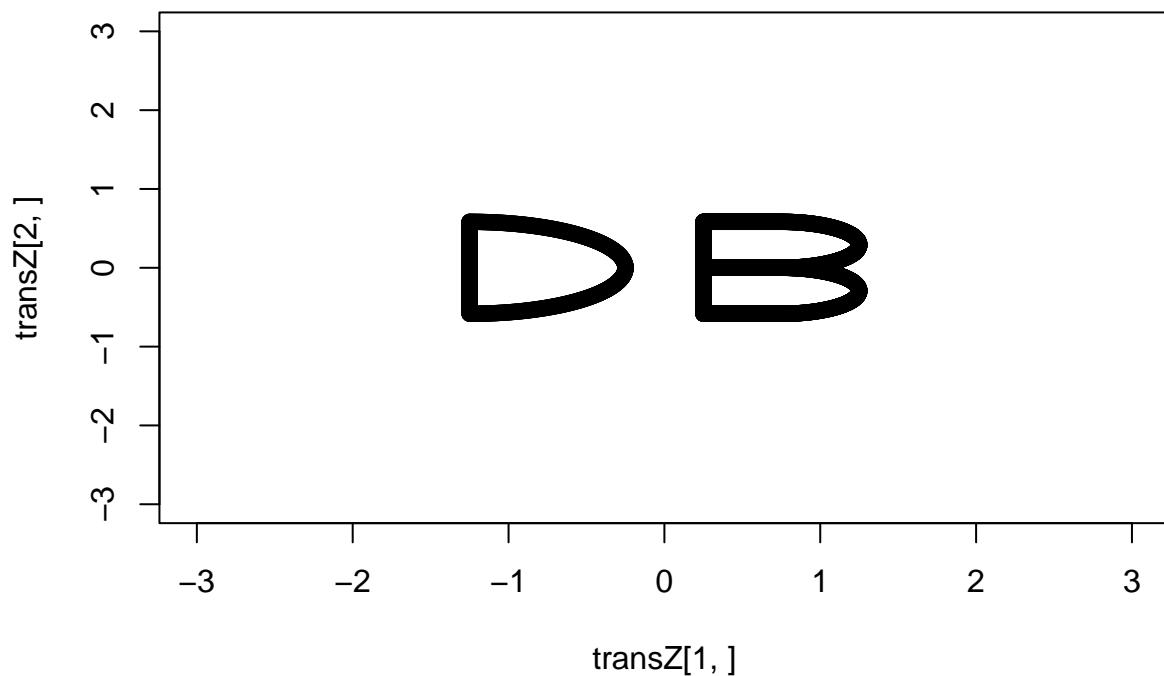


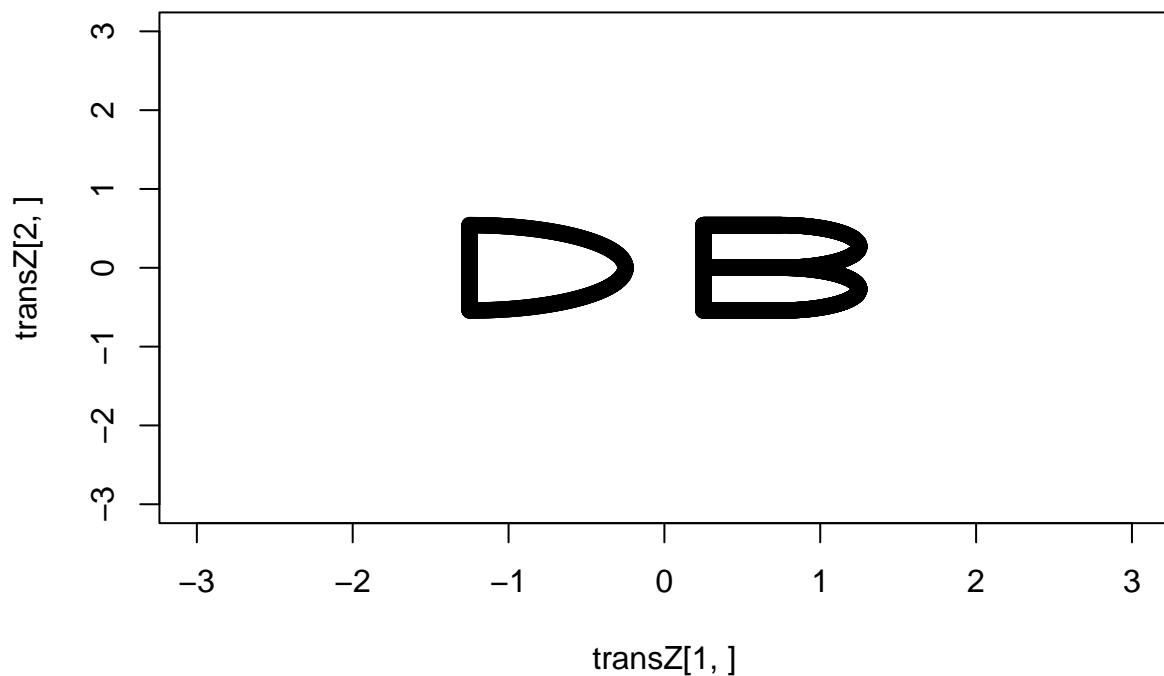


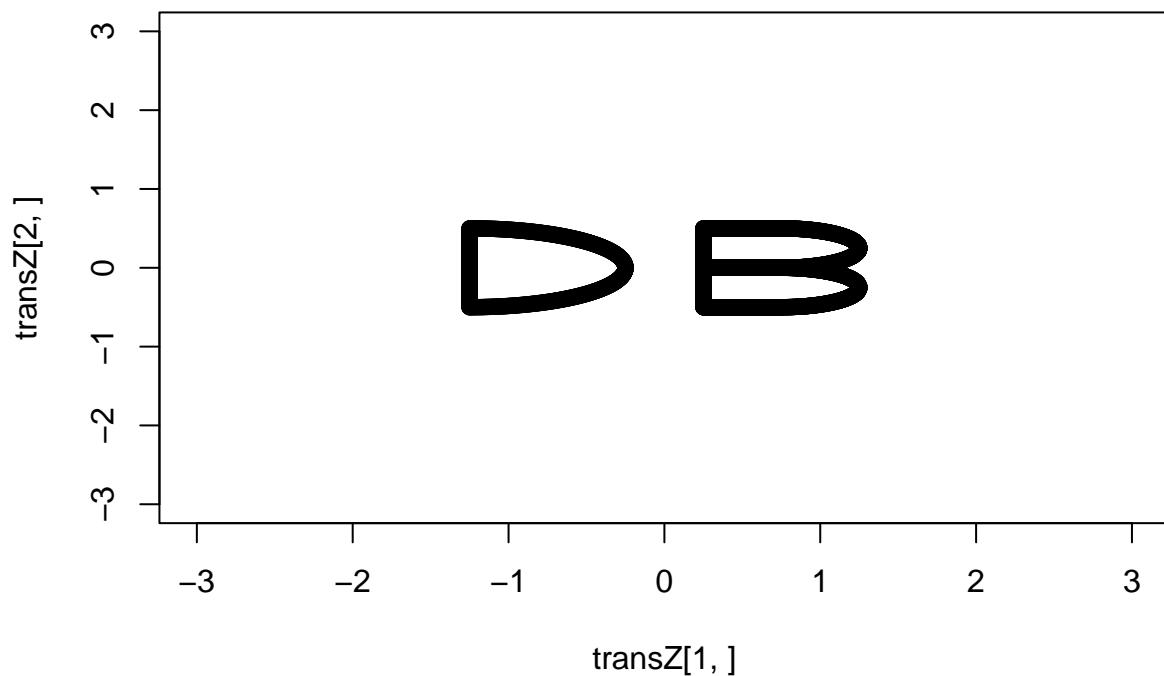


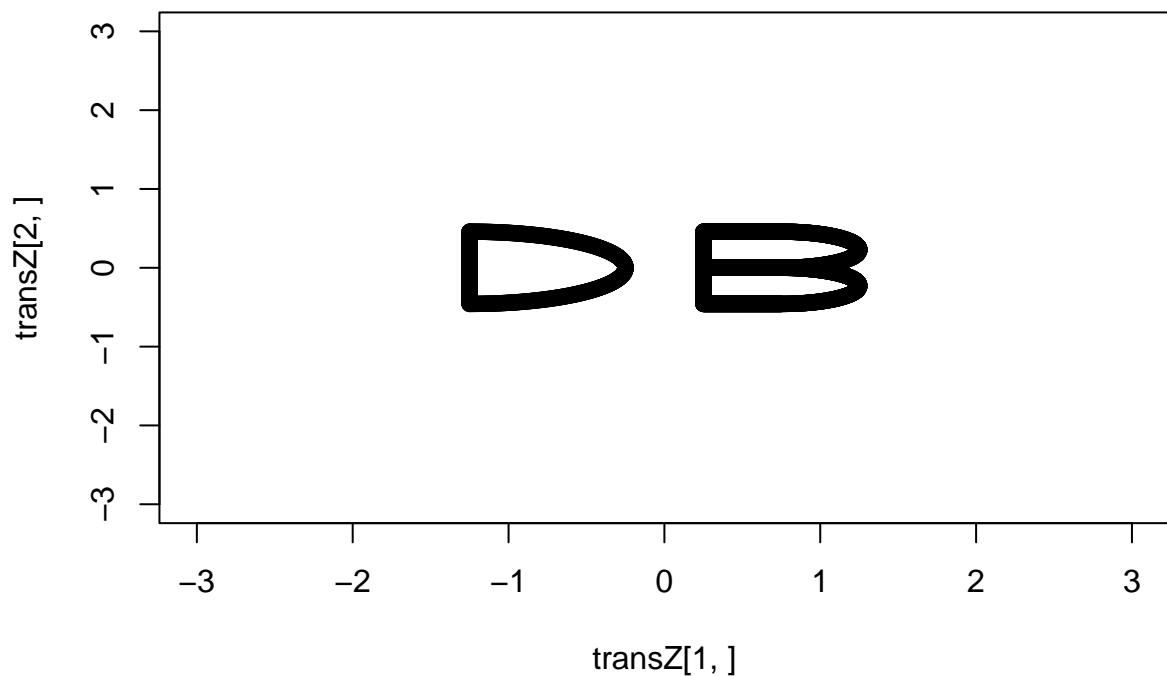


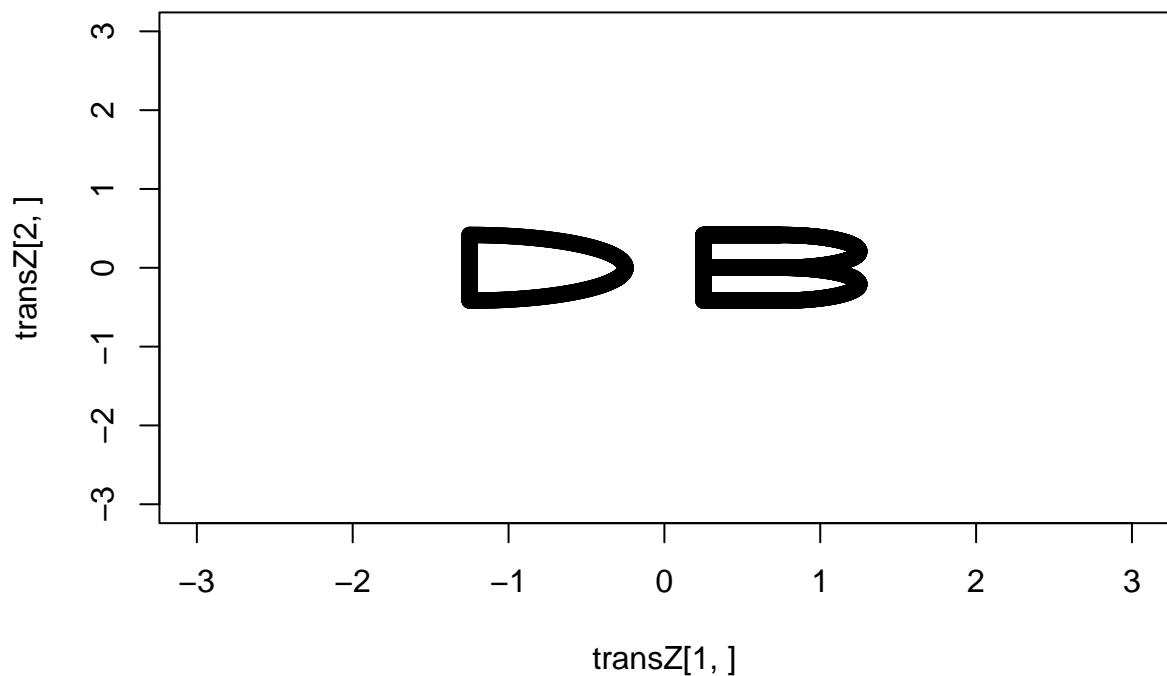


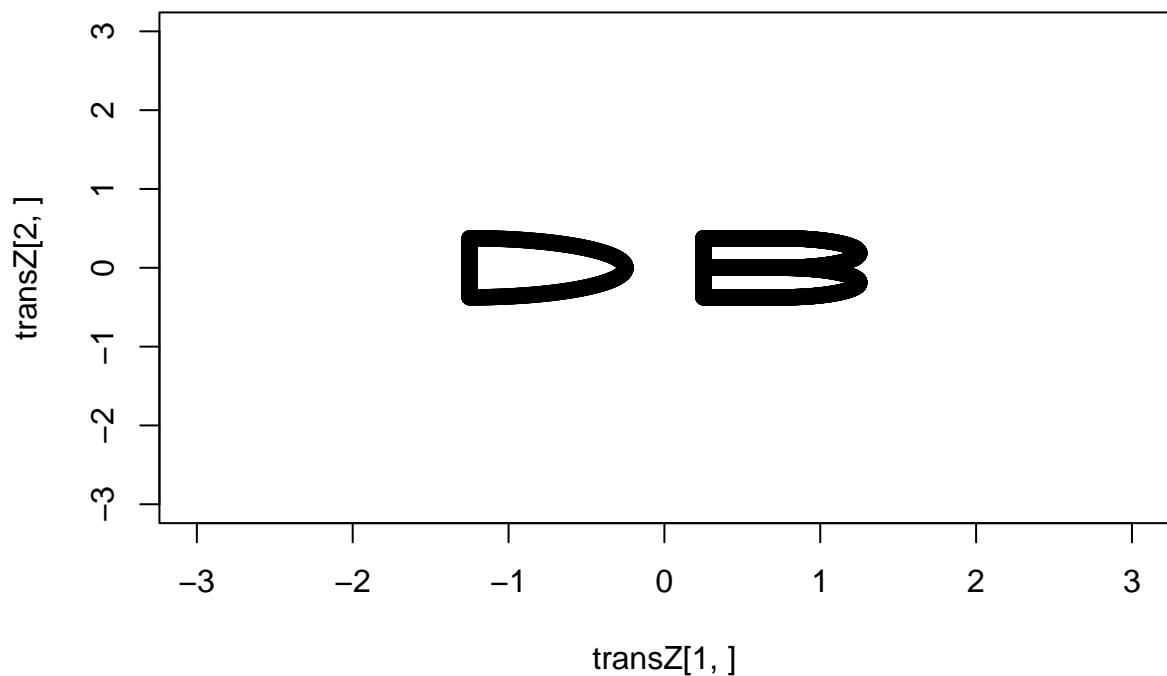


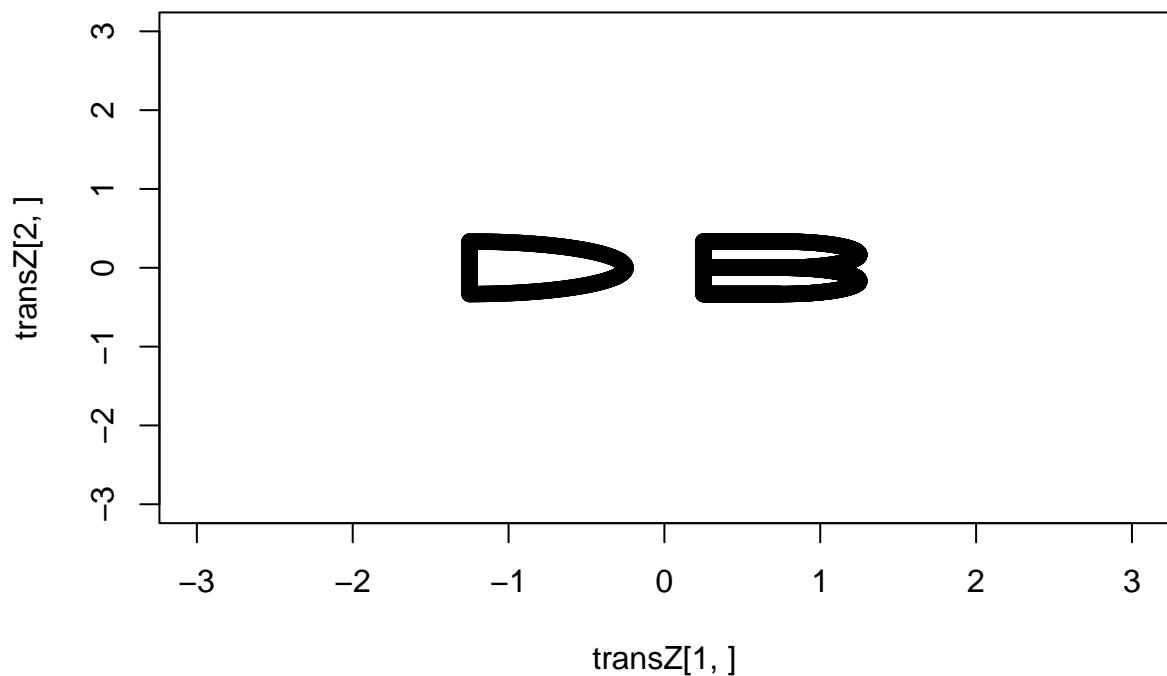


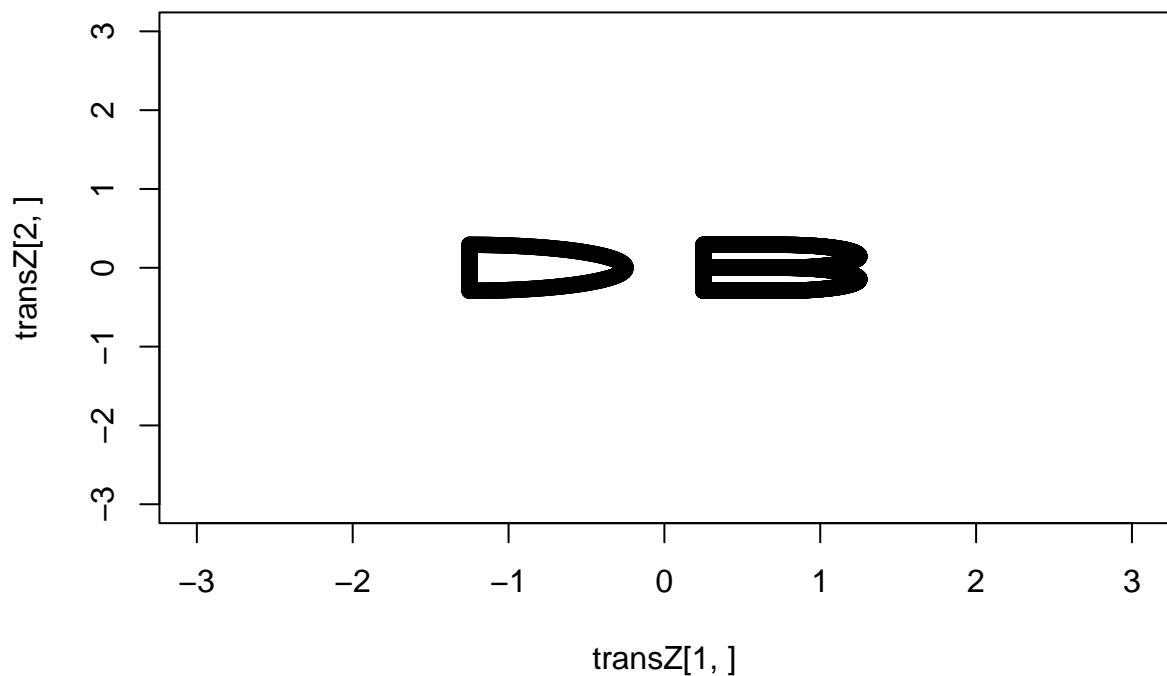


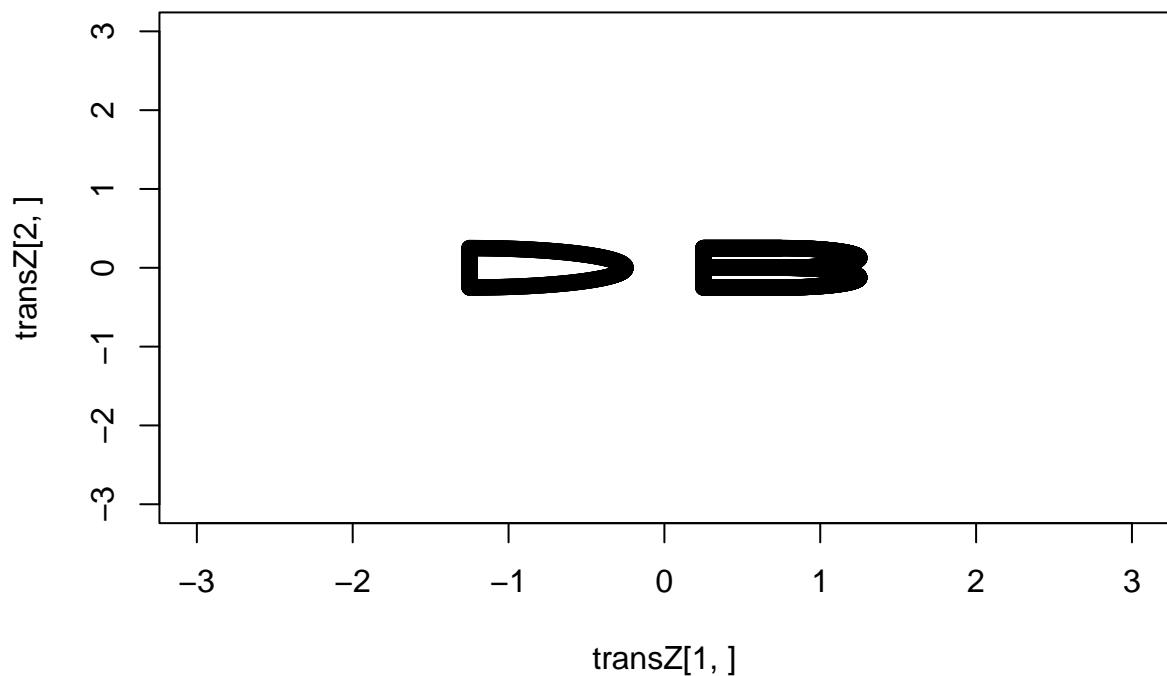


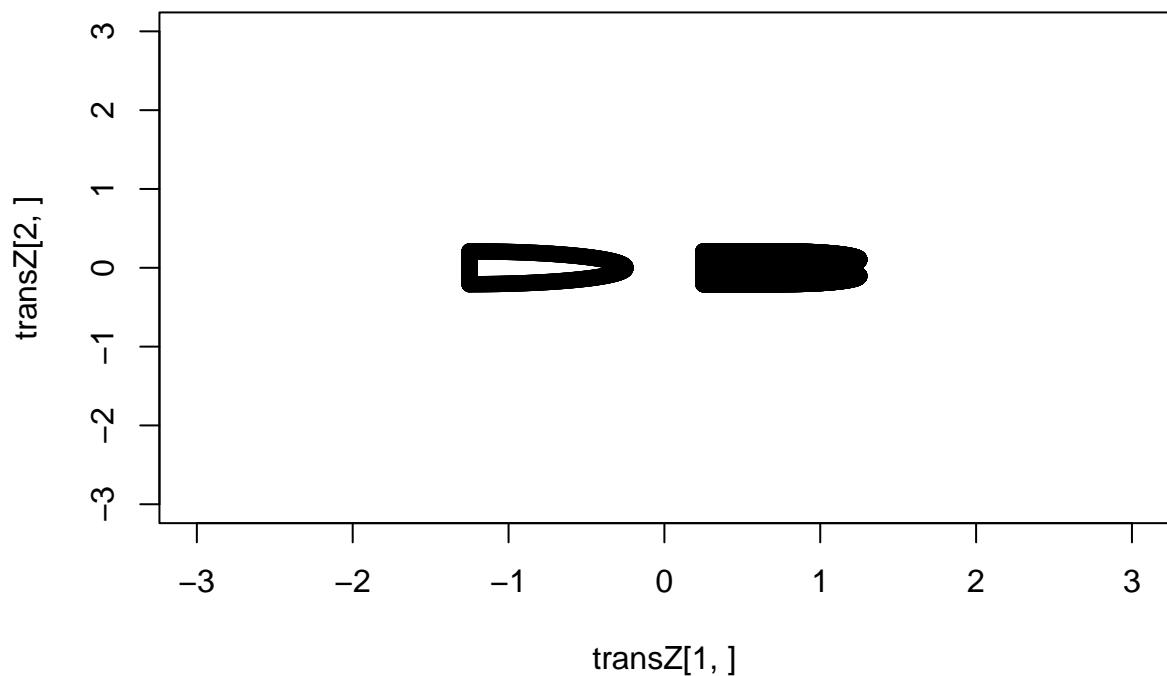


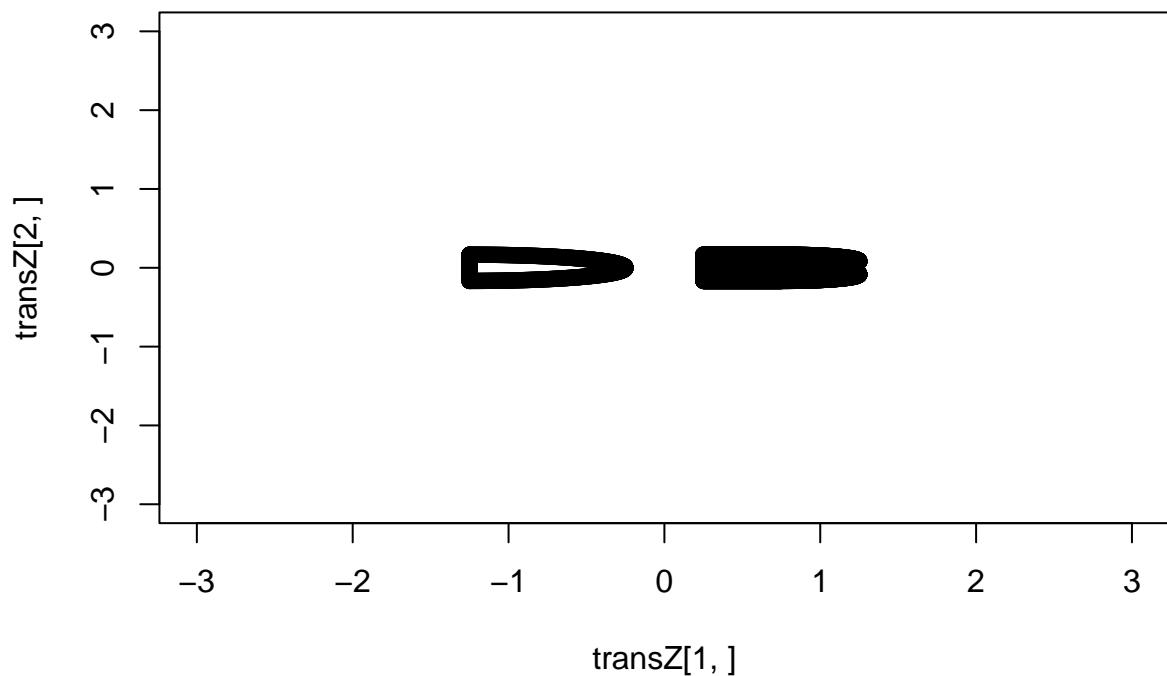


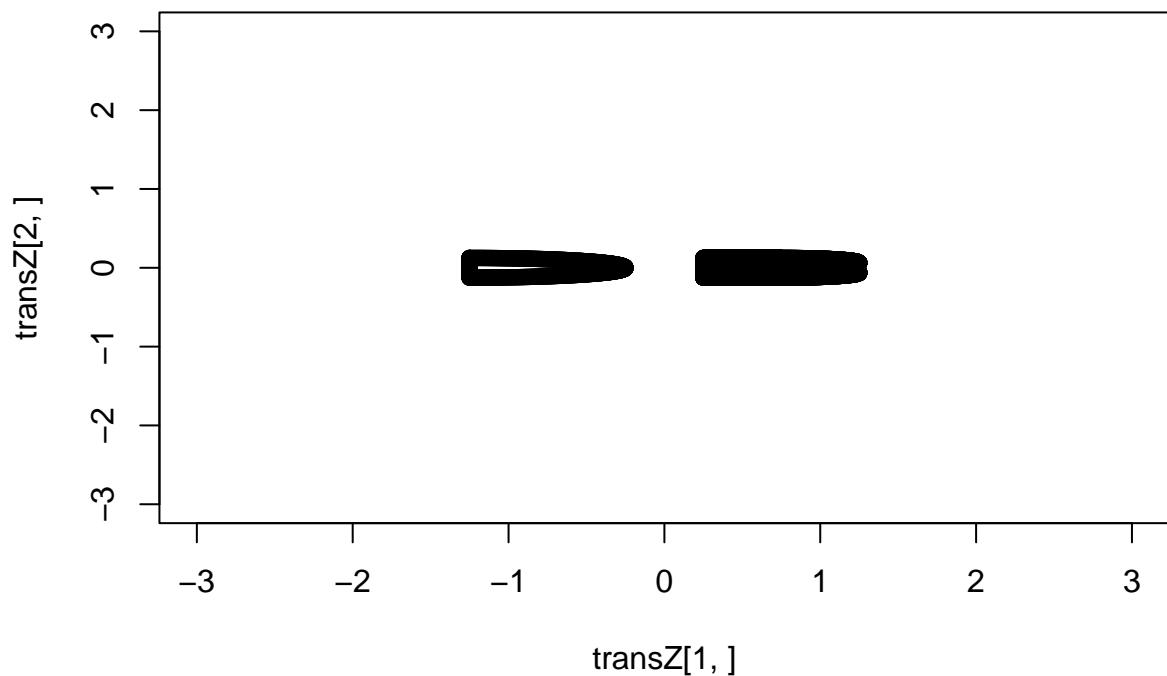


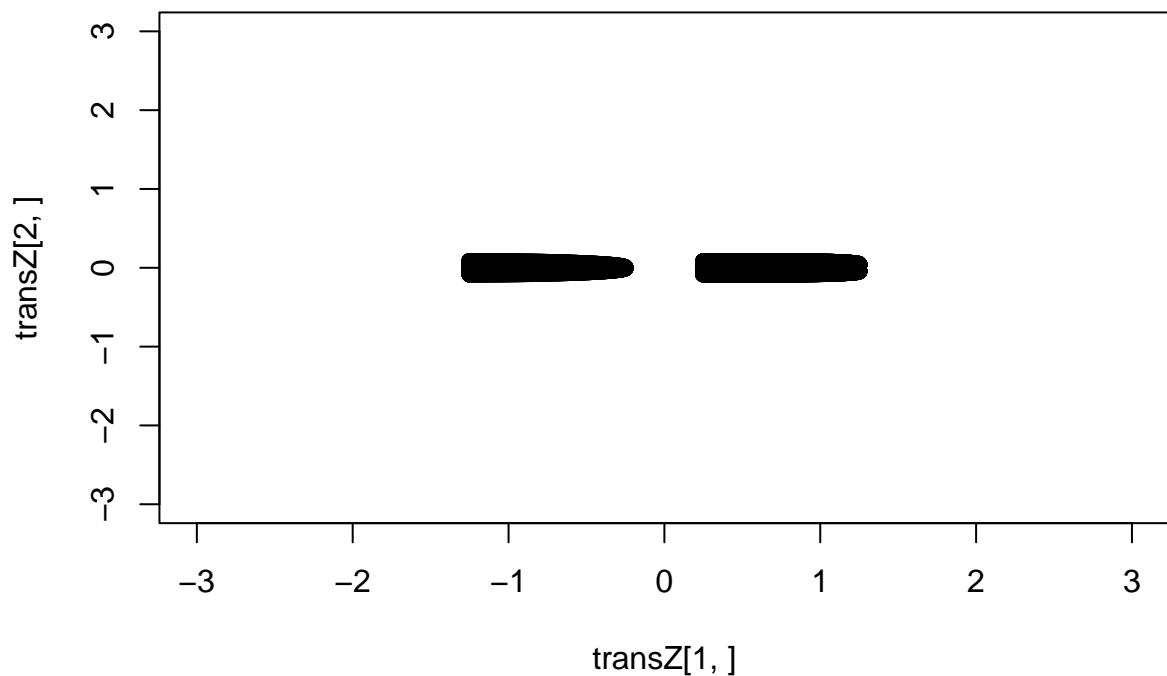


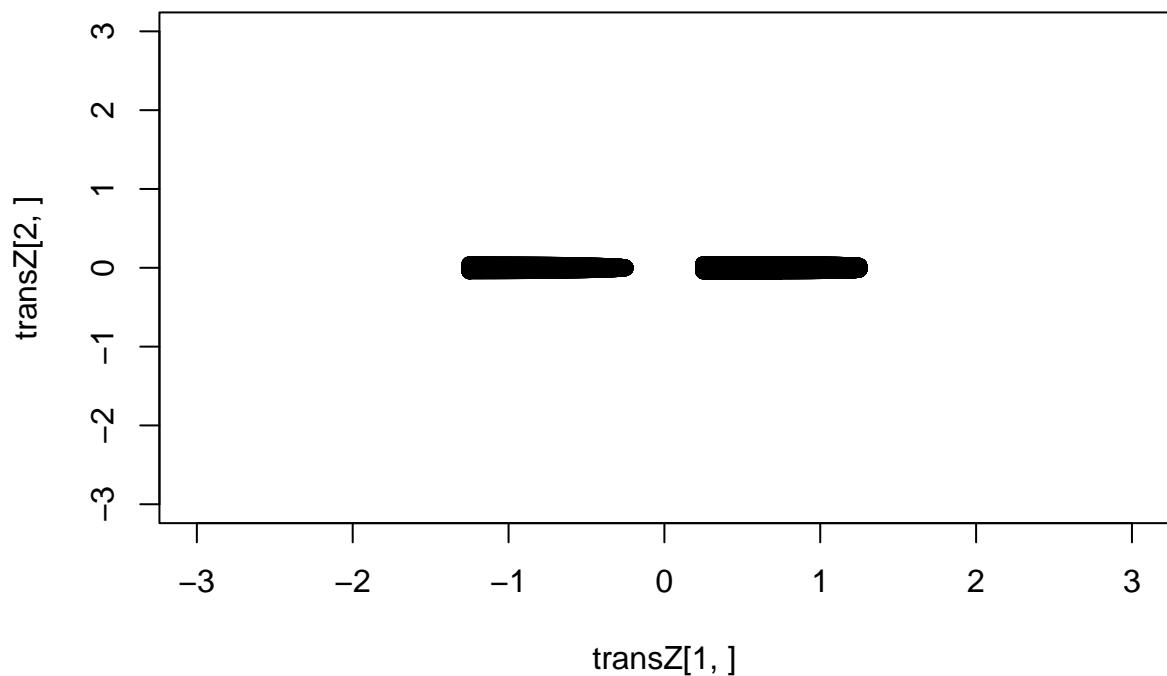


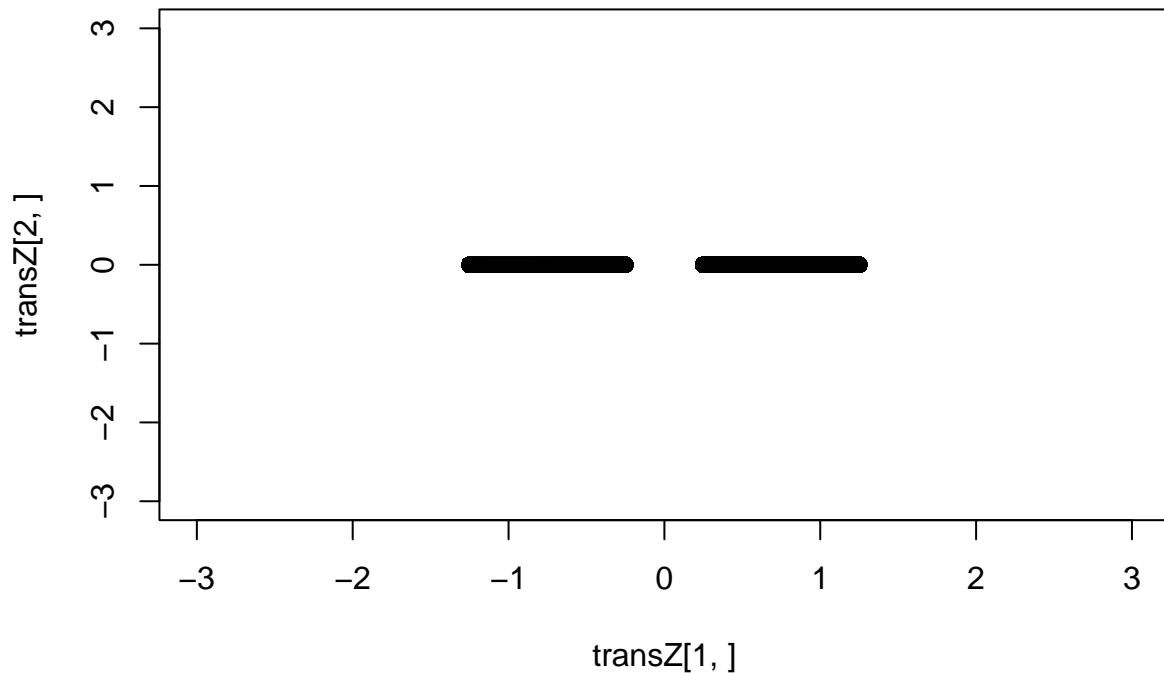






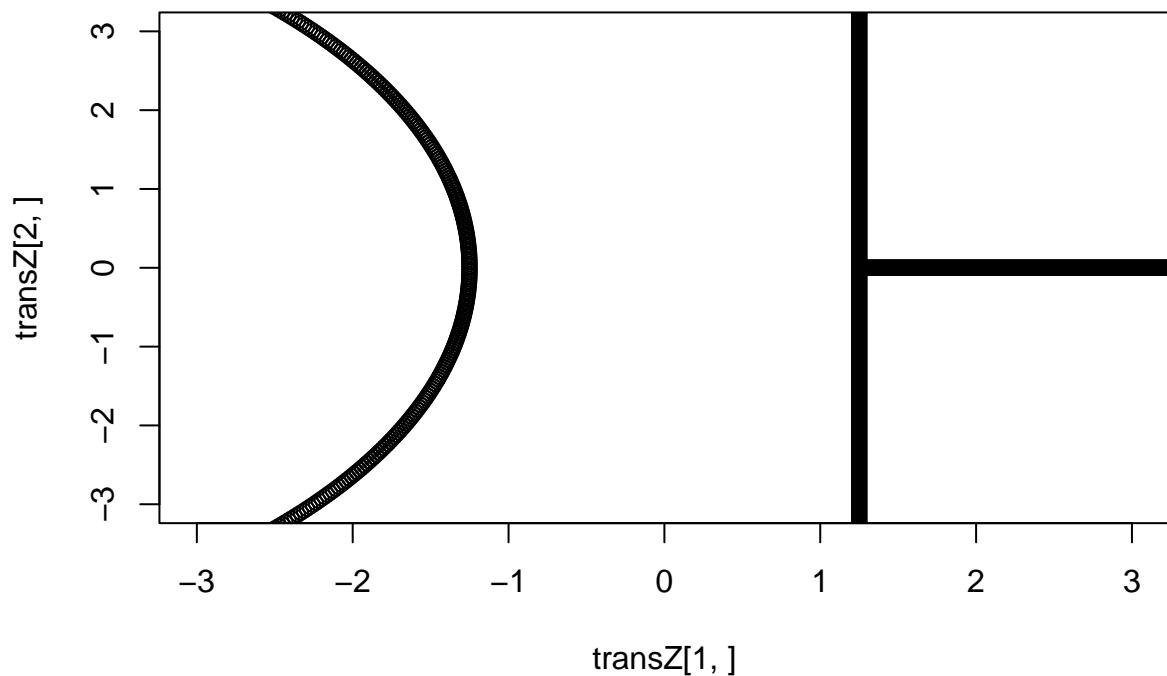


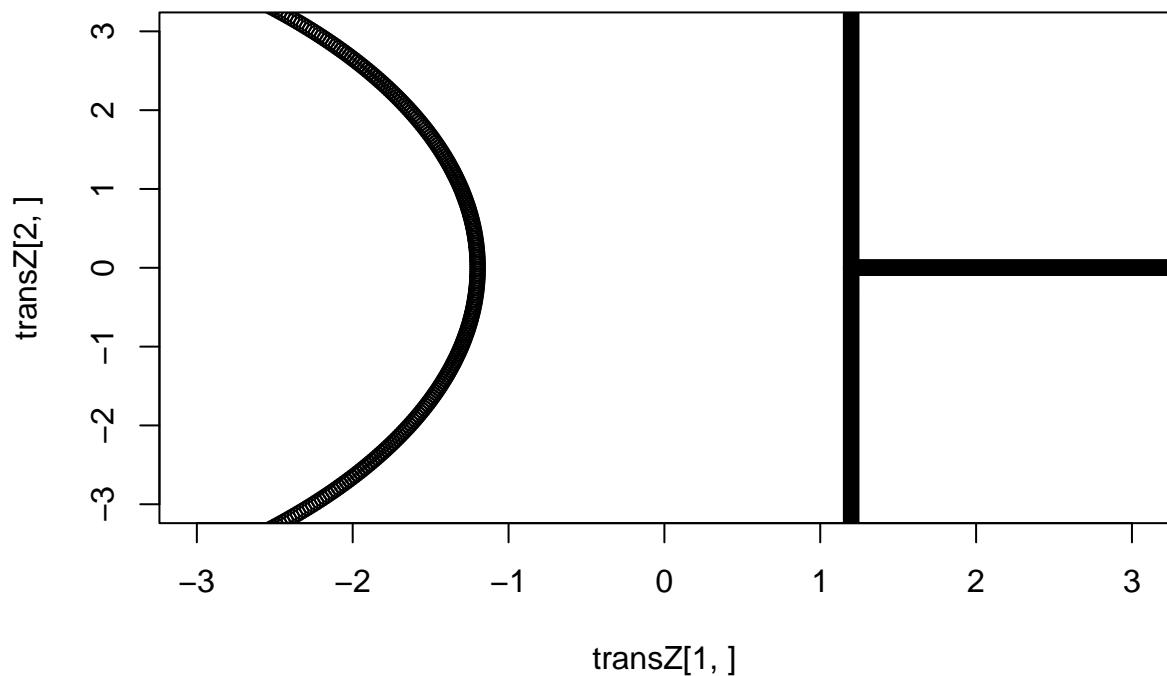


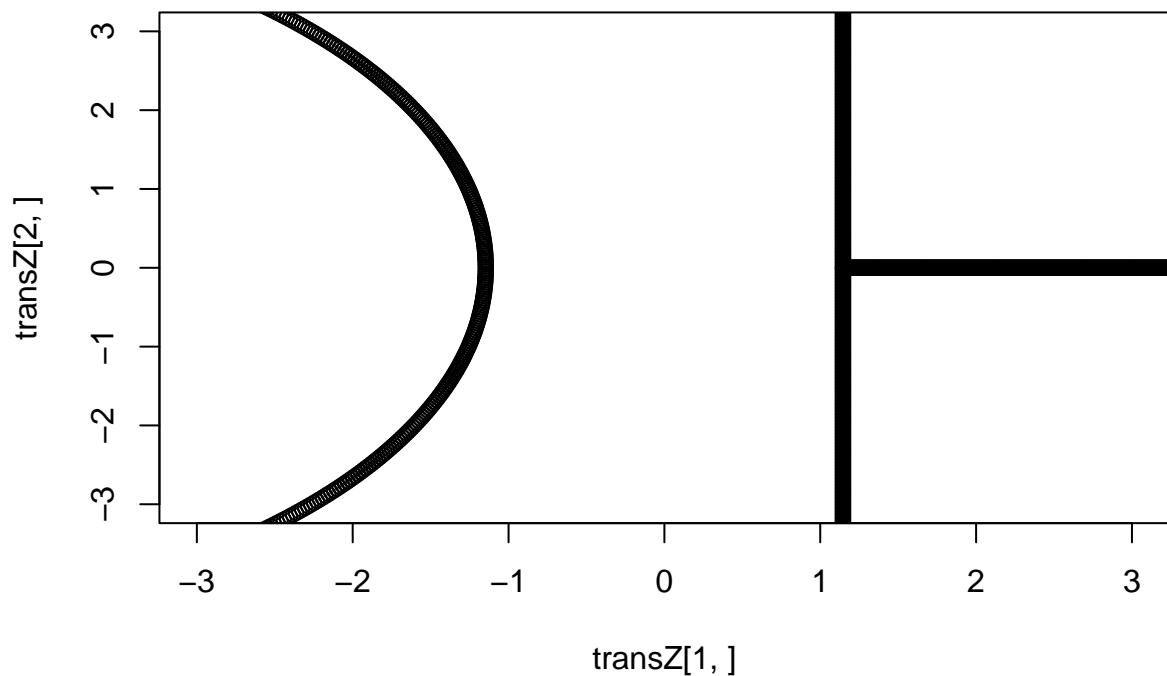


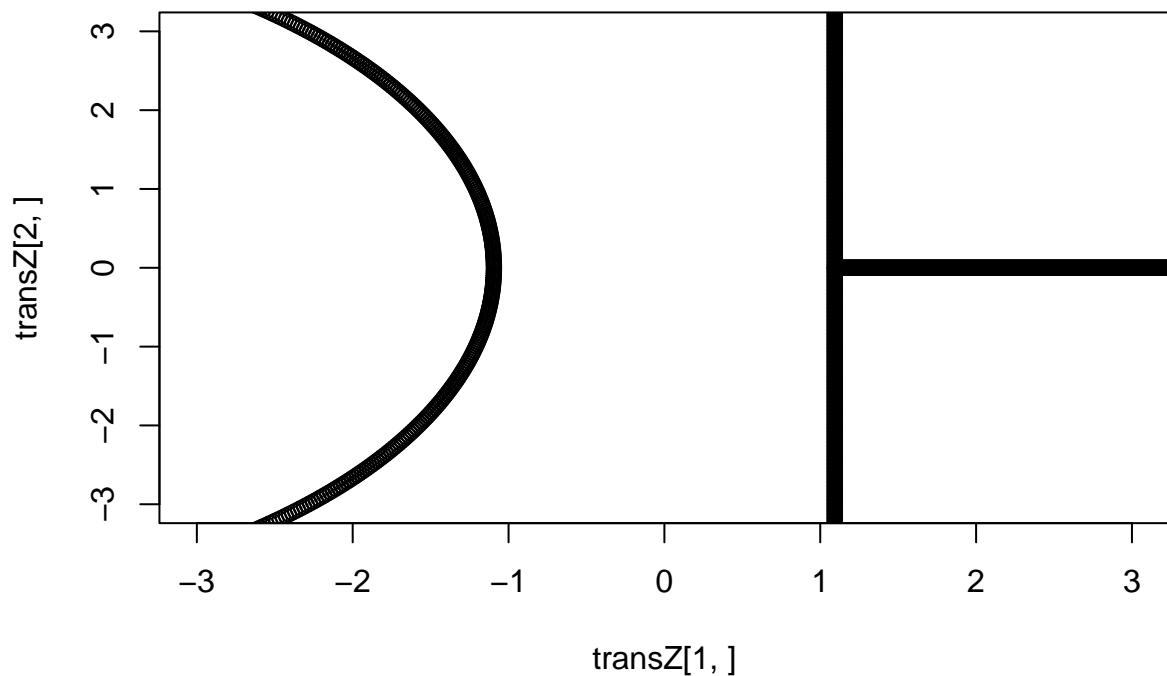
Scaling Uniformly both directions

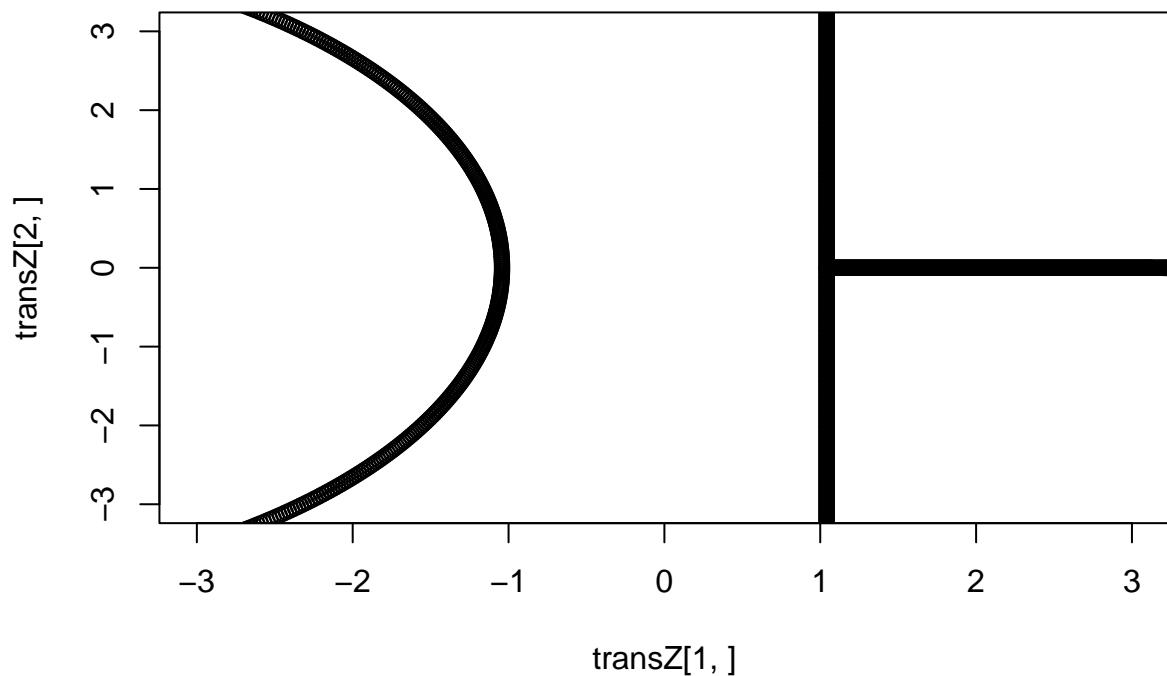
```
x11()
for (scaling_factor in seq(5,0,length.out=25)) {
  transZ <- apply(z,2,function(x) leftMultiply(x,matrix(c(scaling_factor,0,0,scaling_factor),nrow = 2, ncol = 2)))
  plot(transZ[2,]~transZ[1,], xlim = c(-3,3), ylim = c(-3,3))
}
```

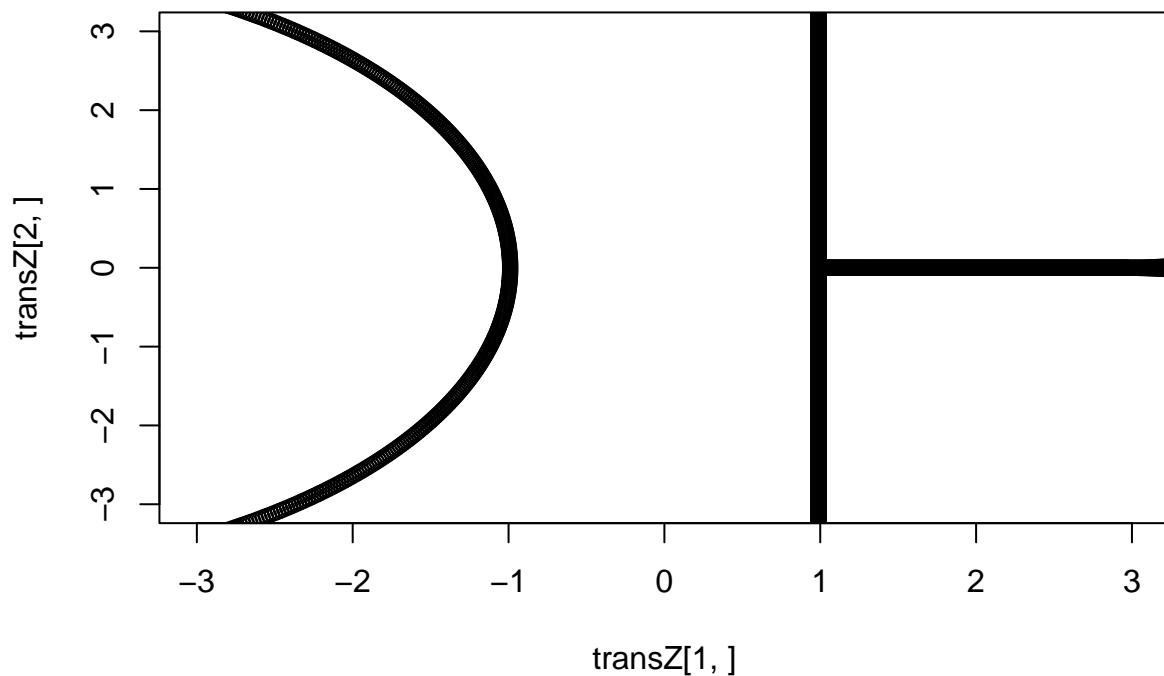


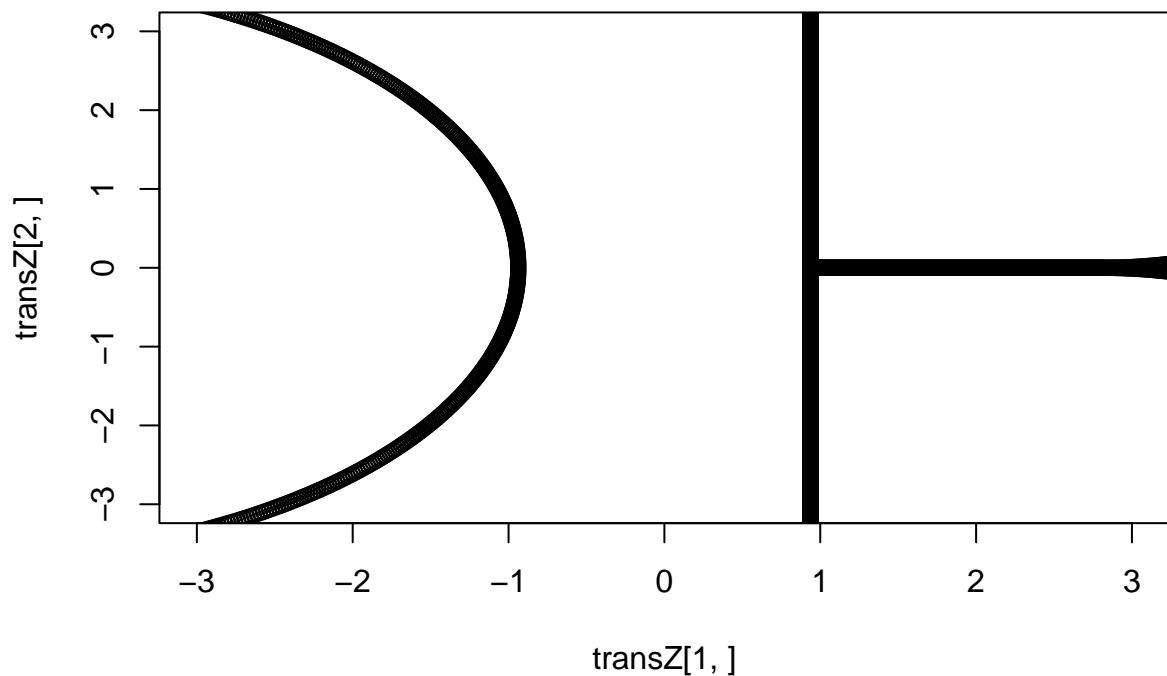


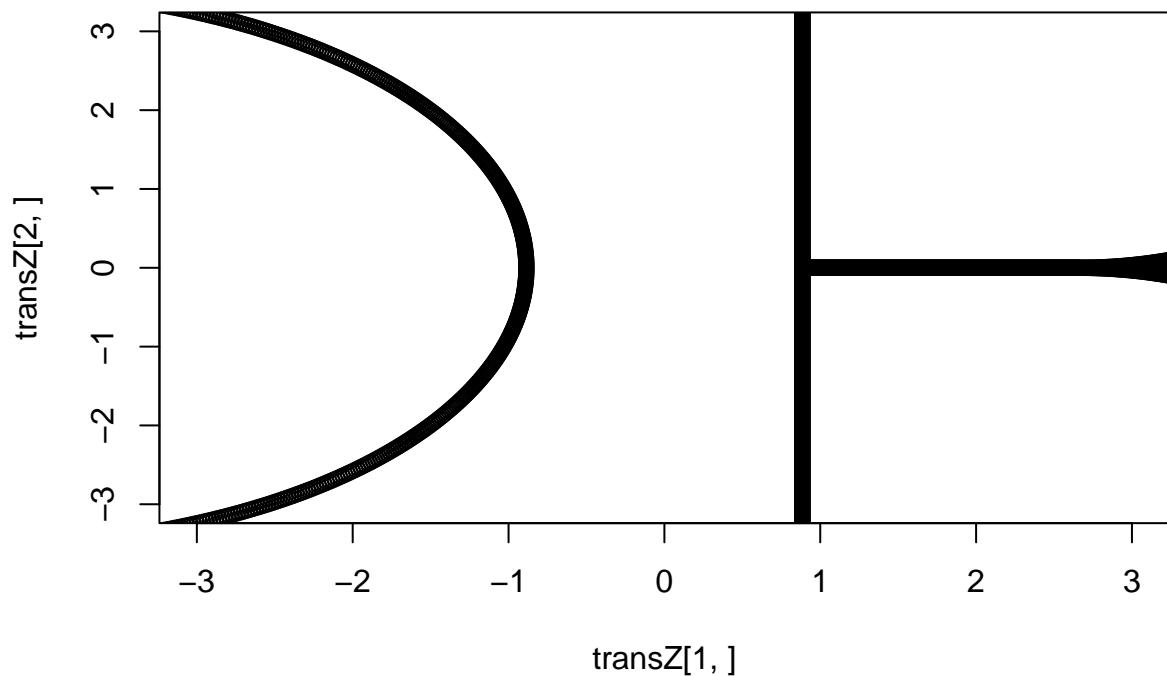


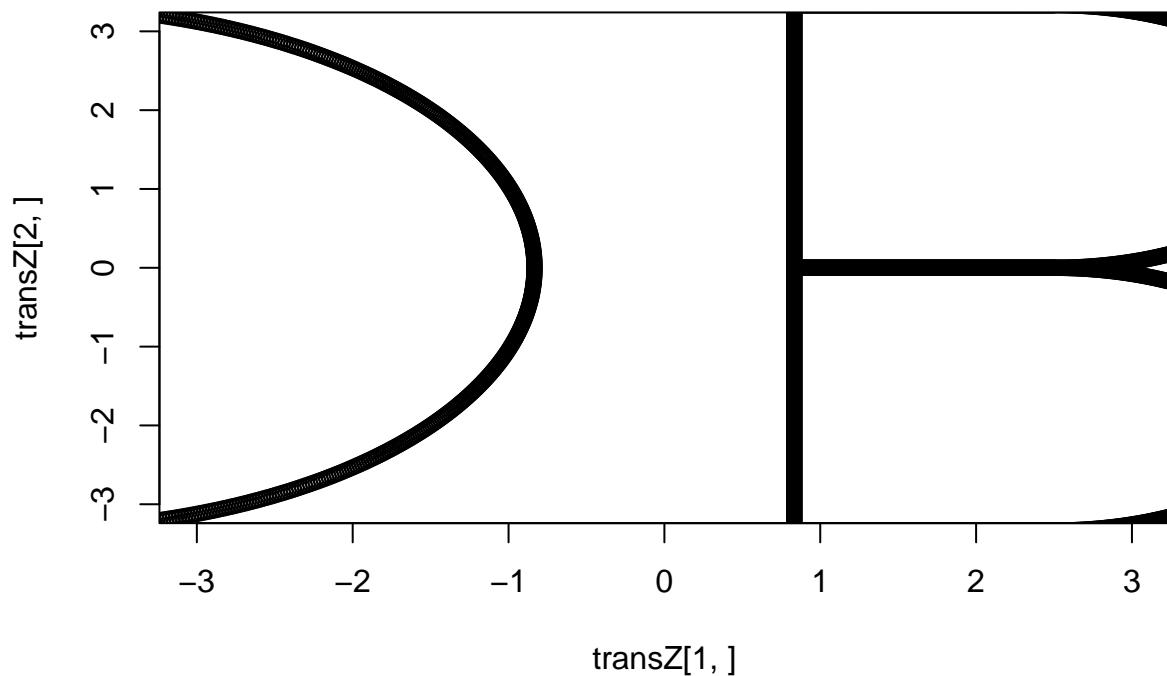


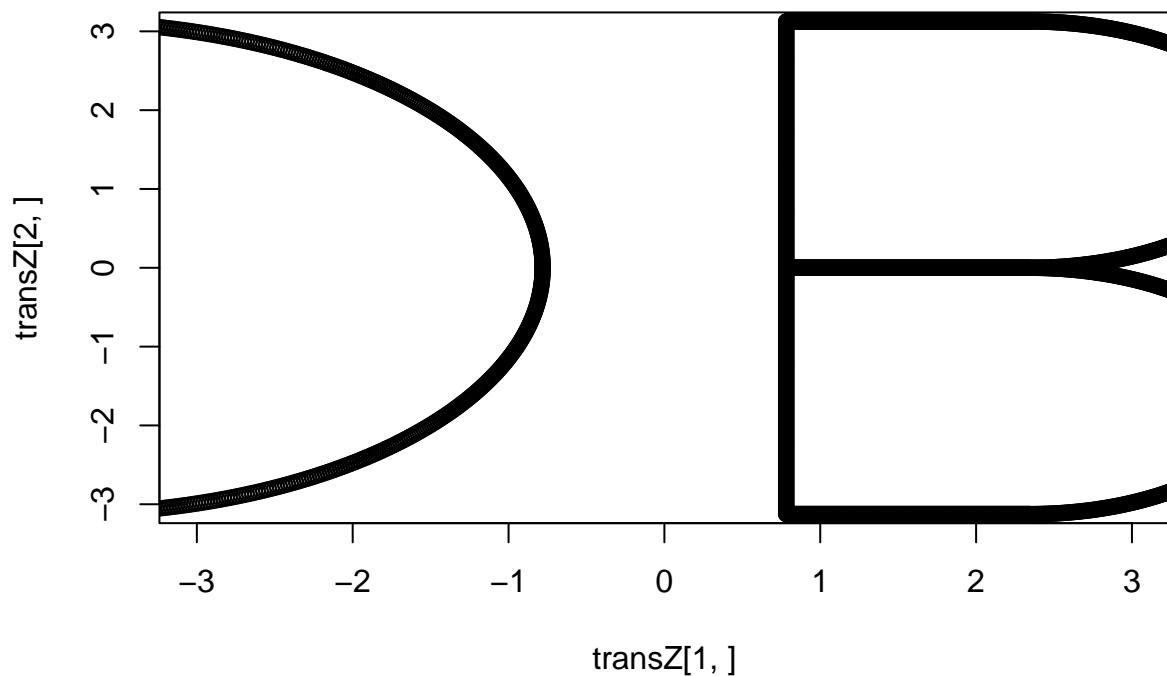


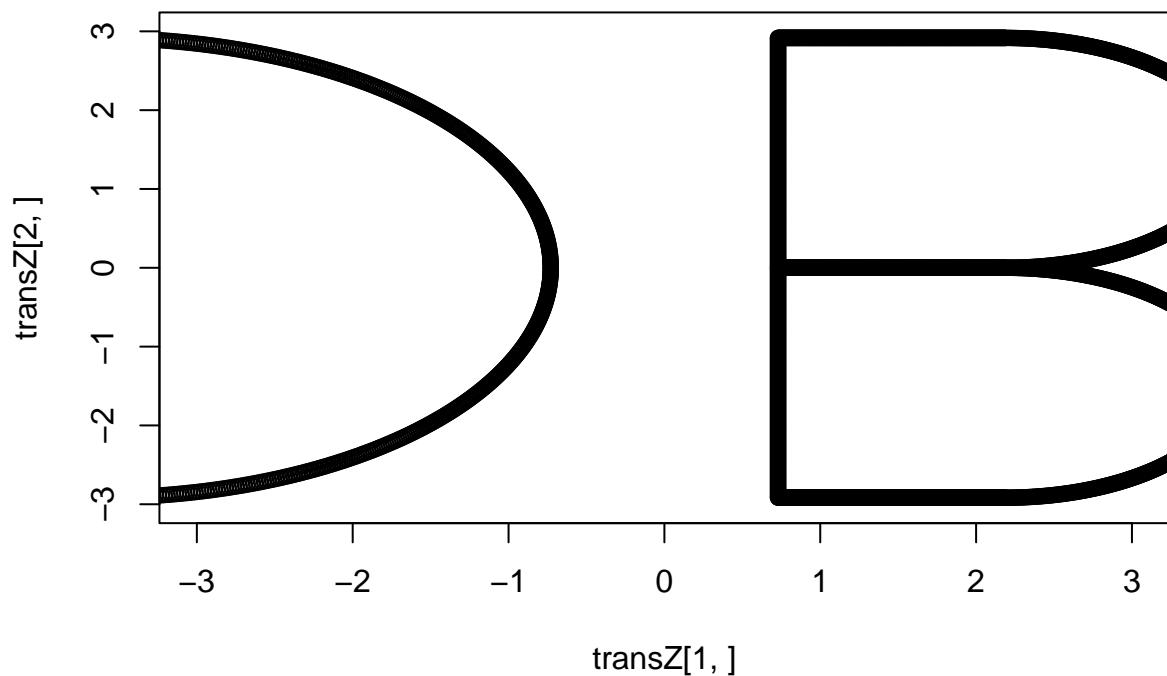


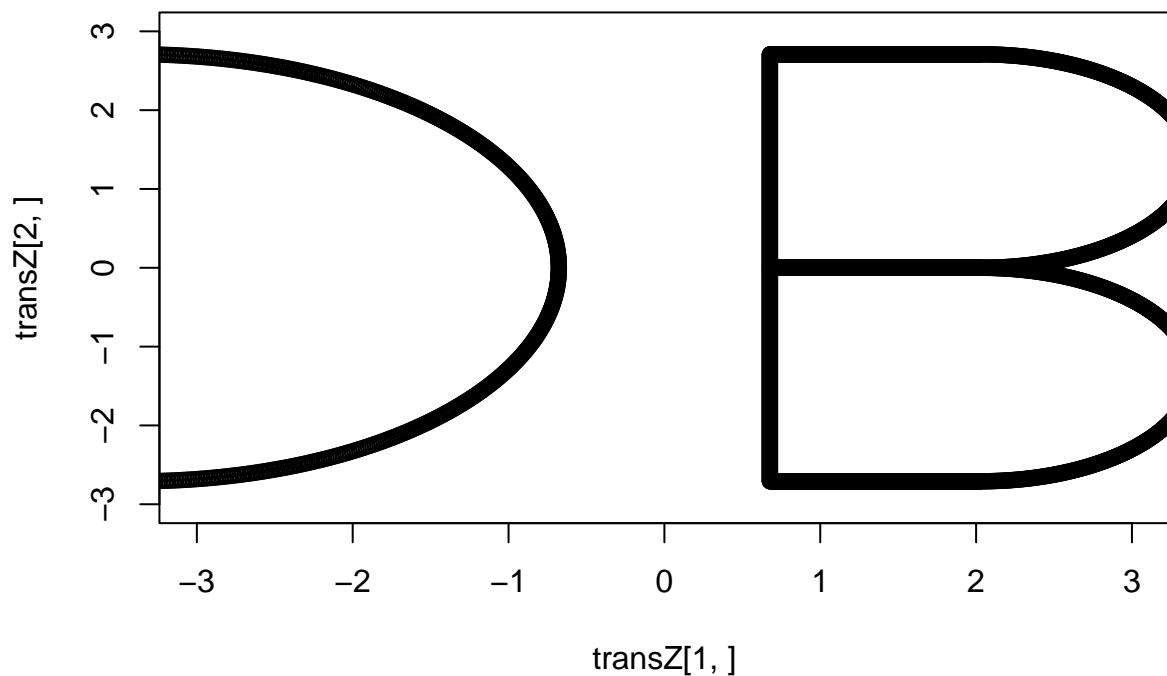


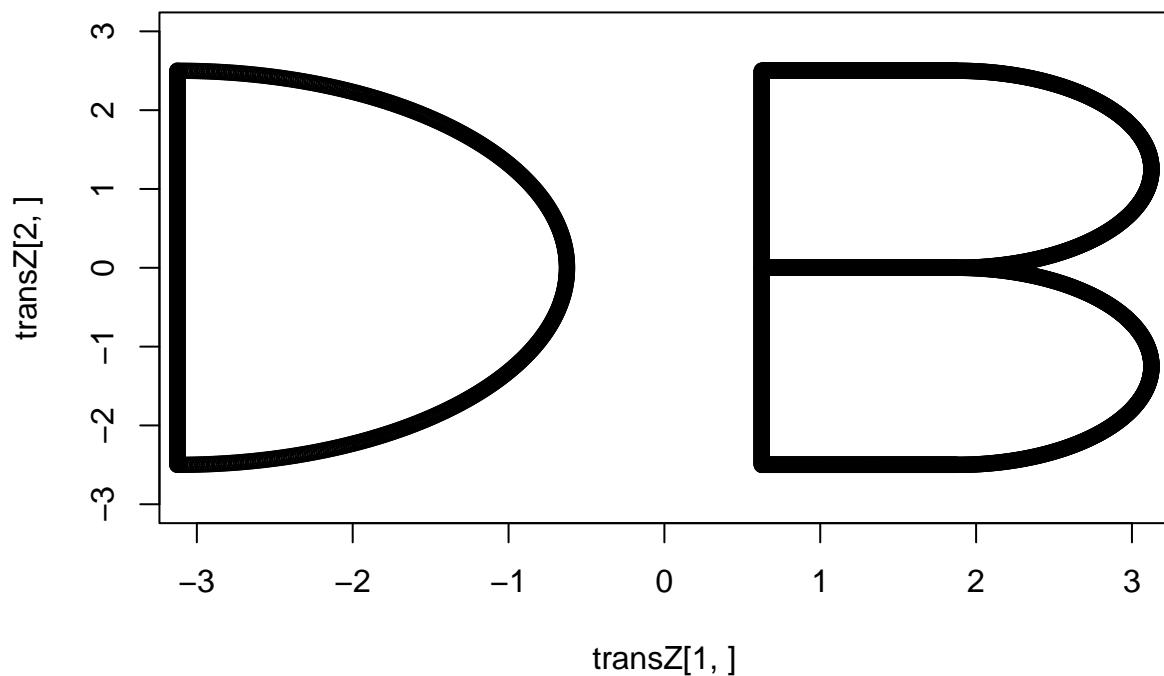


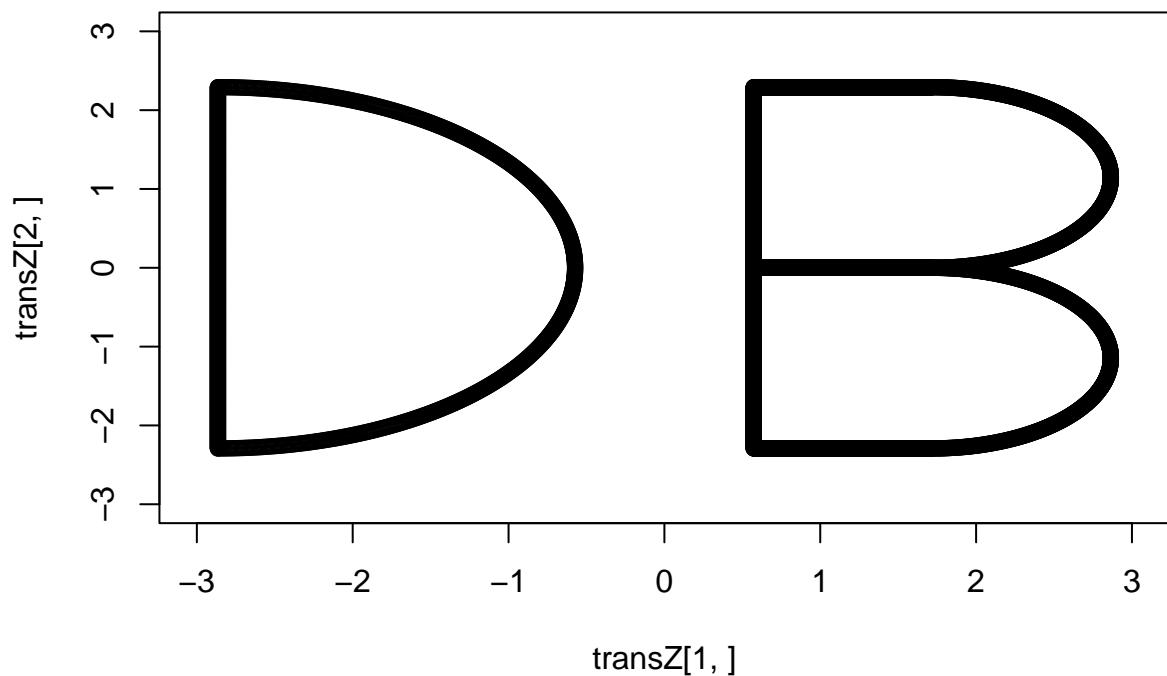


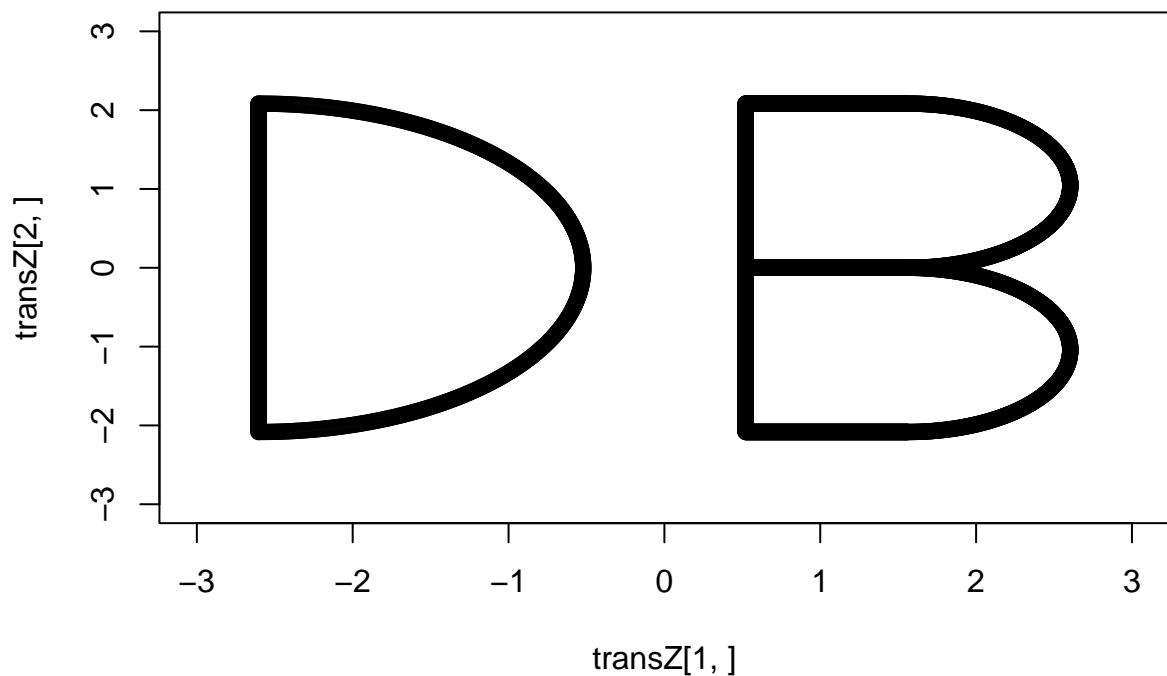


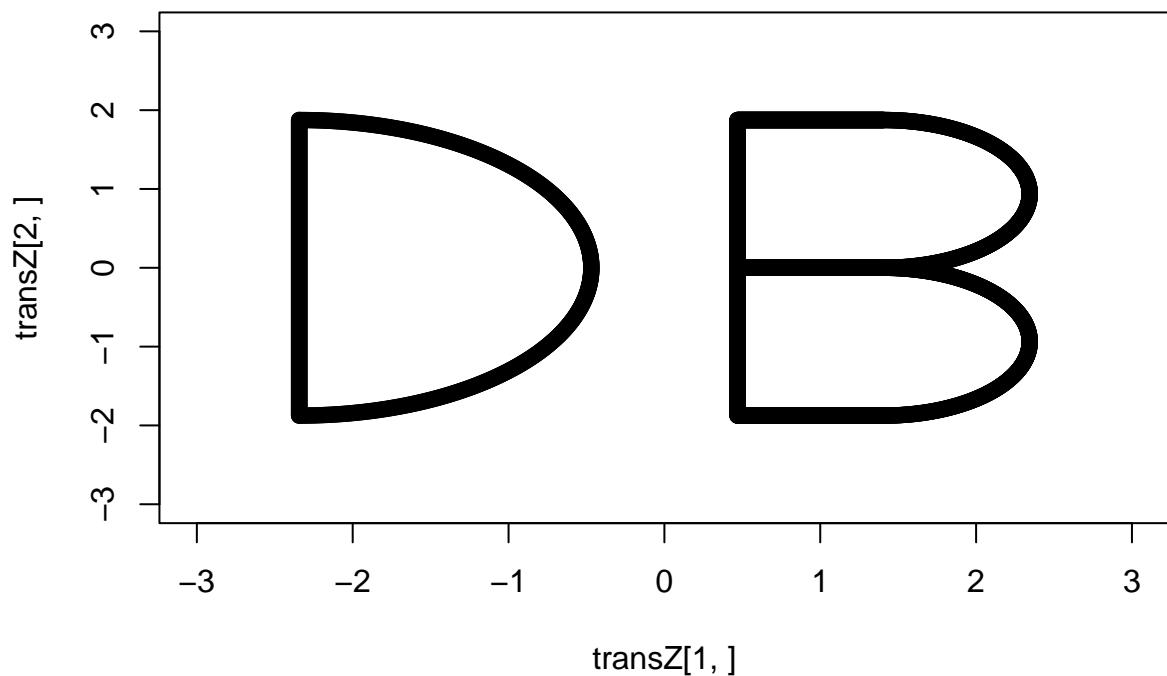


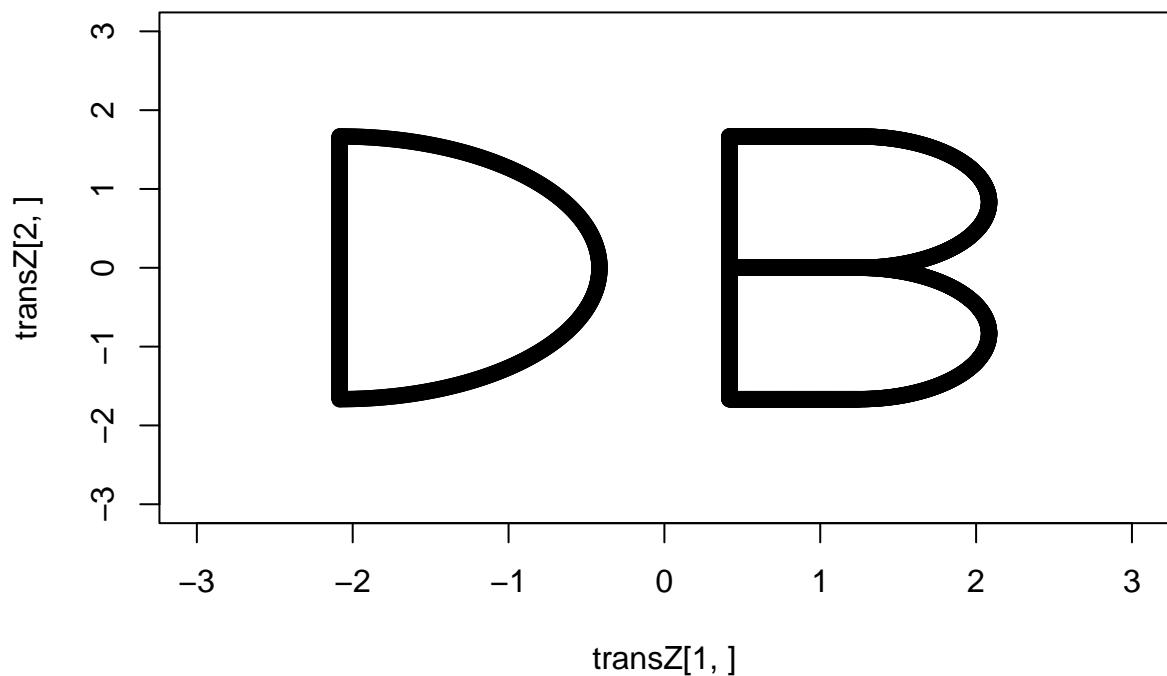


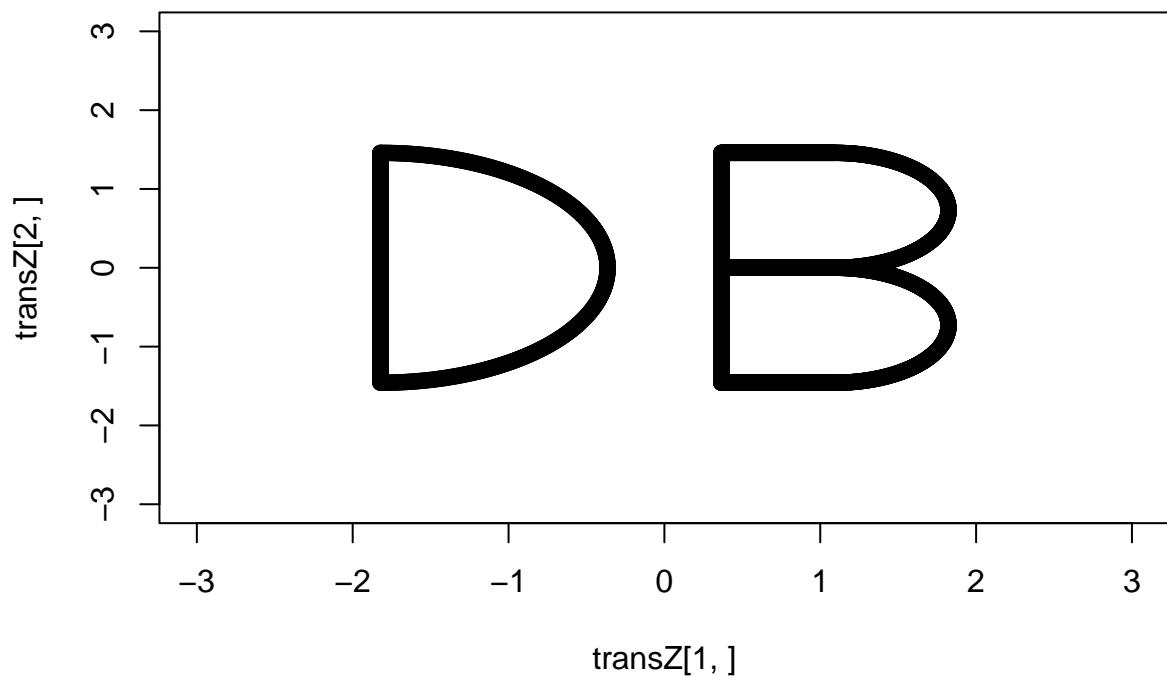


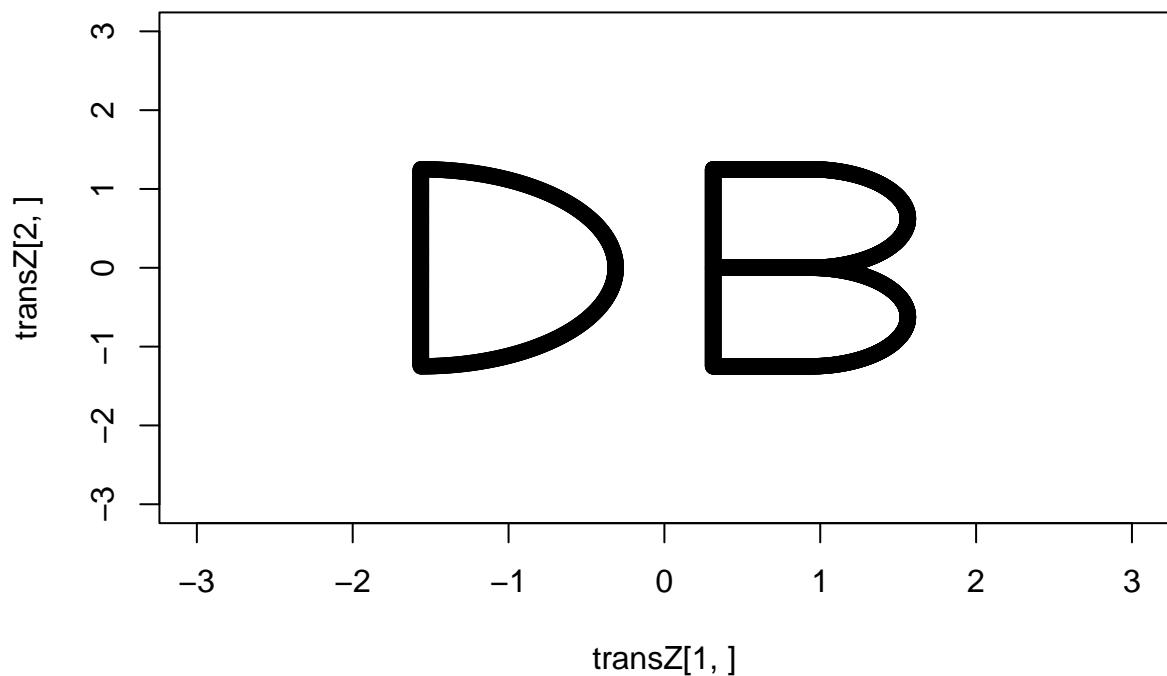


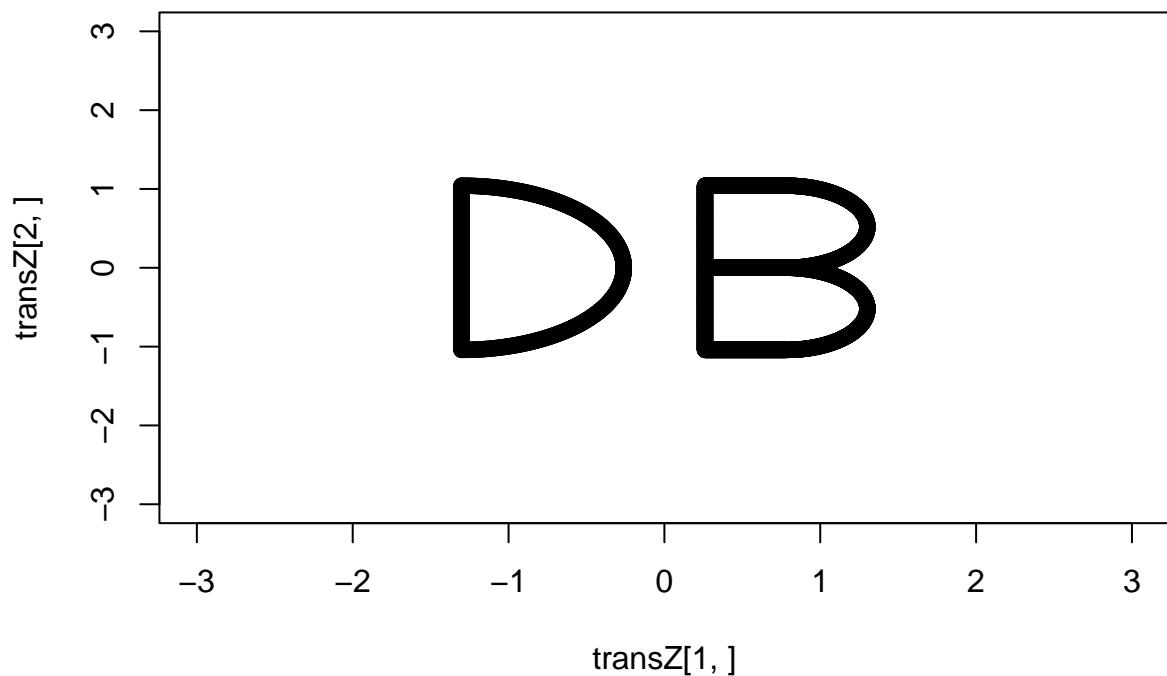


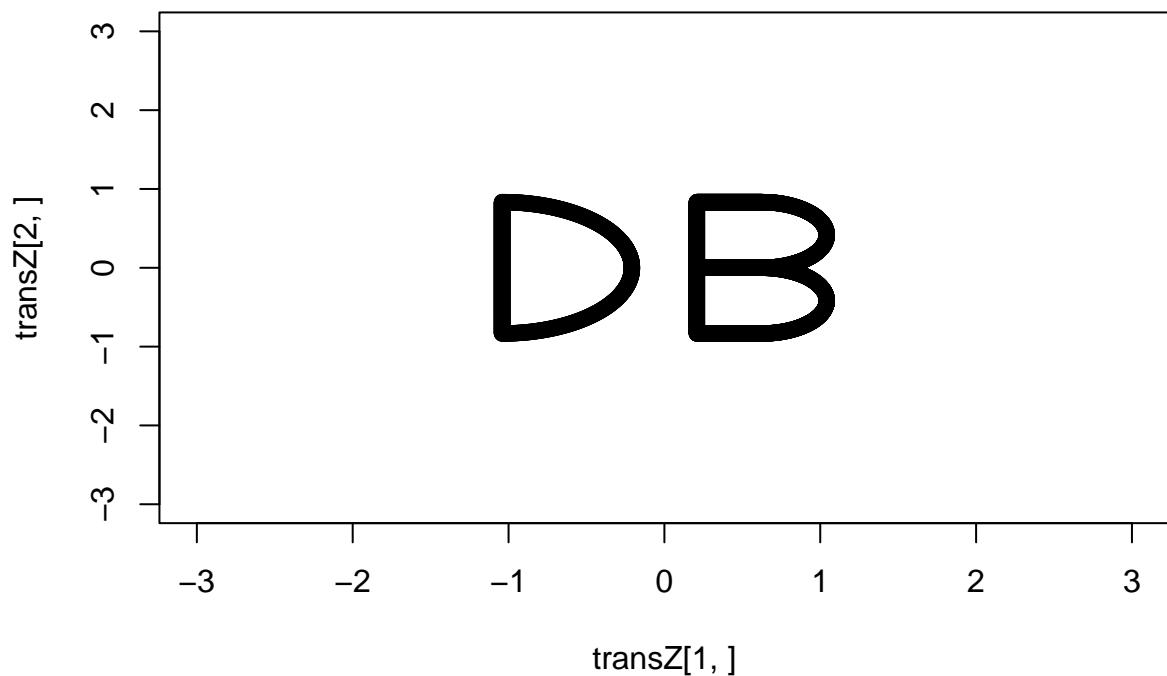


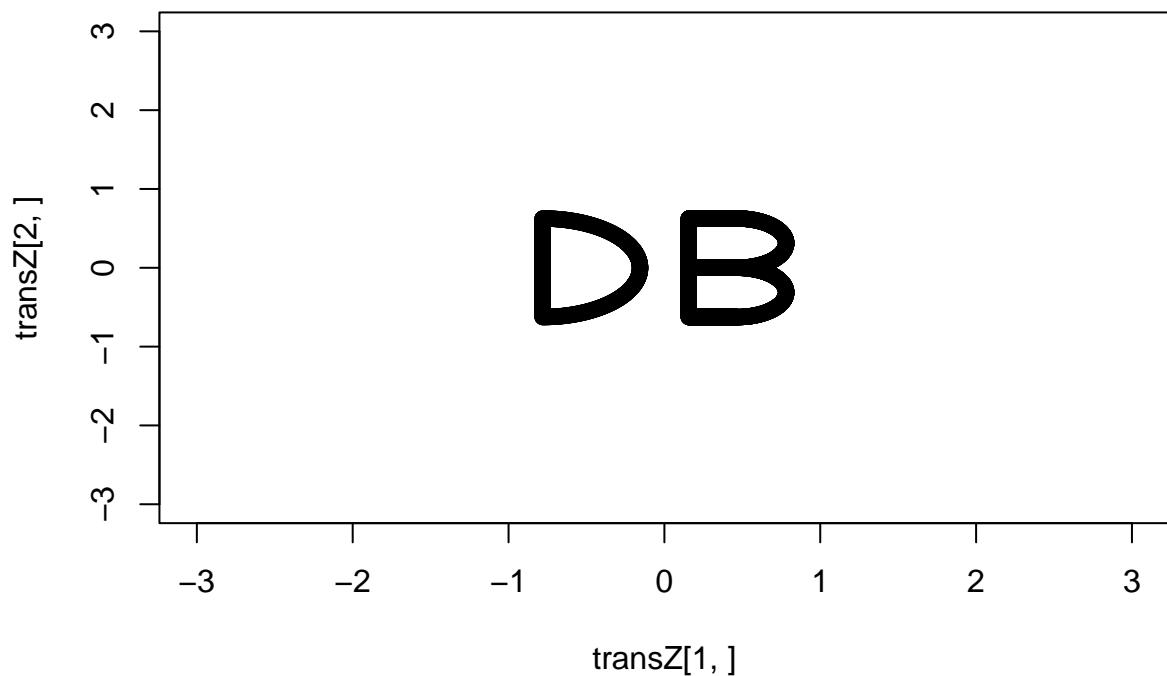


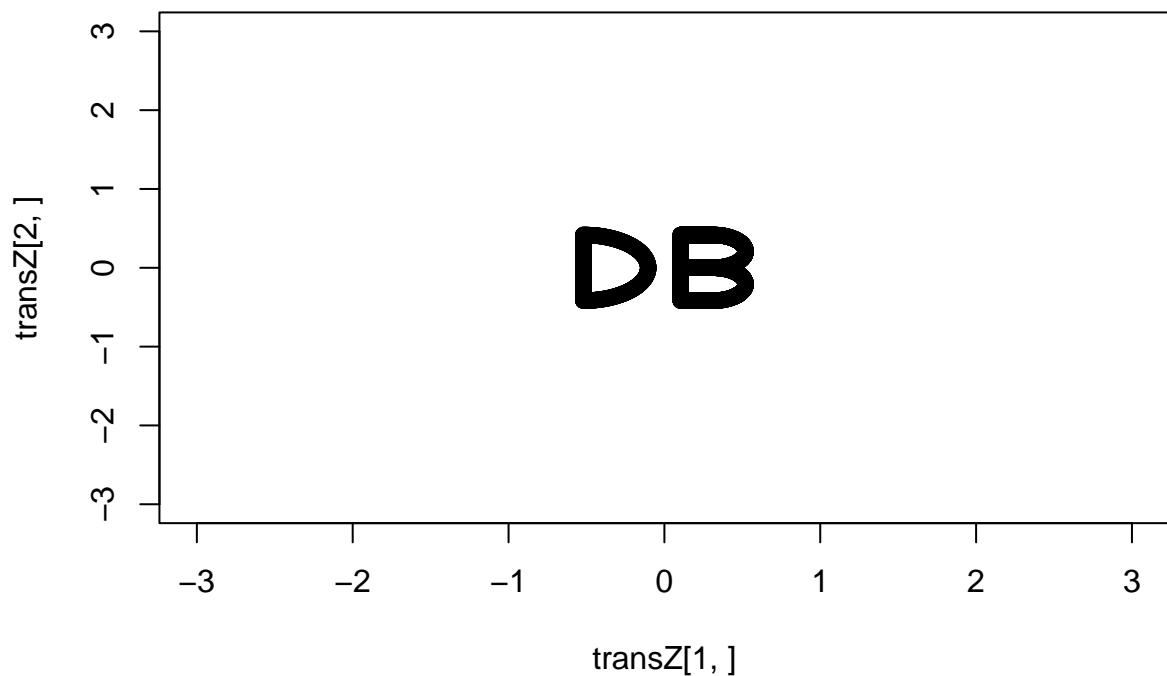




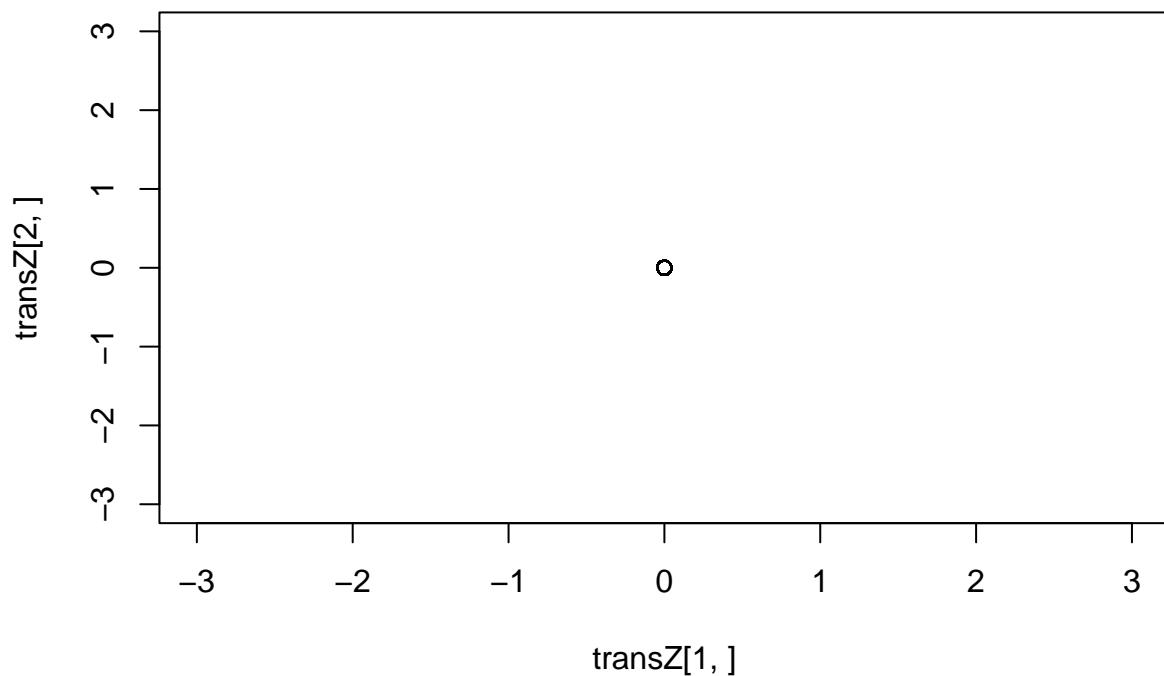












## Rotation

The rotation mapping applies trigonometric functions to rotate around the origin. The mapping for angle

$$\theta$$

is