script—Copie–2-.R

mosta

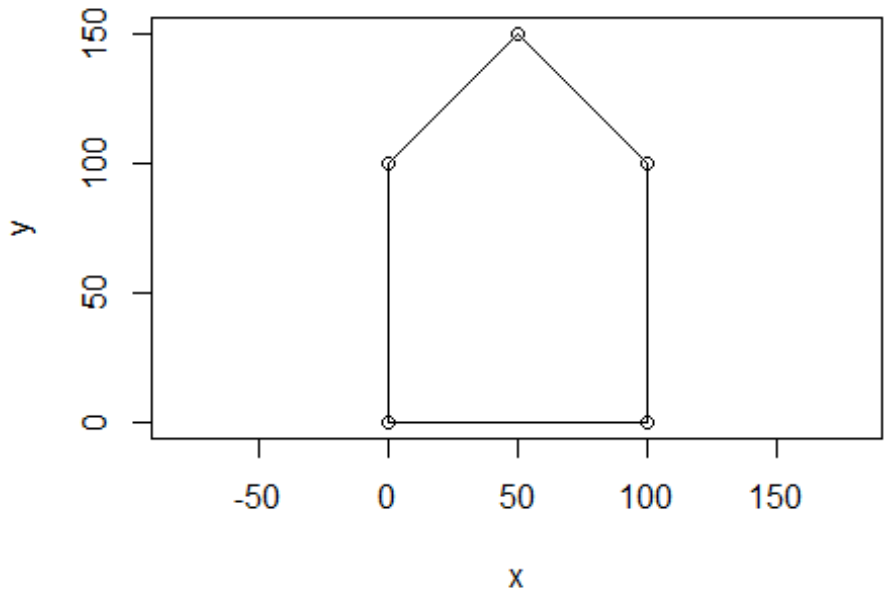
2021-05-28

```
library(RJSONIO)

## Warning: package 'RJSONIO' was built under R version 4.0.3

library(lme4)

## Warning: package 'lme4' was built under R version 4.0.3

## Loading required package: Matrix

library(lmerTest)

## Warning: package 'lmerTest' was built under R version 4.0.3

##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.5

library (mlmRev)

## Warning: package 'mlmRev' was built under R version 4.0.3

library(rjson)

## Warning: package 'rjson' was built under R version 4.0.3

##
## Attaching package: 'rjson'

## The following objects are masked from 'package:RJSONIO':
##
##     fromJSON, toJSON

library(jsonlite)
```

```
## Warning: package 'jsonlite' was built under R version 4.0.3

##
## Attaching package: 'jsonlite'

## The following objects are masked from 'package:rjson':
##
##     fromJSON, toJSON

## The following objects are masked from 'package:RJSONIO':
##
##     fromJSON, toJSON

library (foreign)
library(curl)

## Warning: package 'curl' was built under R version 4.0.3

library (mlmRev)
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v tibble  3.0.4     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.3

## Warning: package 'purrr' was built under R version 4.0.3

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.3

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x tidyr::expand()      masks Matrix::expand()
## x dplyr::filter()      masks stats::filter()
## x purrr::flatten()     masks jsonlite::flatten()
## x jsonlite::fromJSON() masks rjson::fromJSON(), RJSONIO::fromJSON()
## x dplyr::lag()         masks stats::lag()
## x tidyr::pack()        masks Matrix::pack()
## x readr::parse_date()  masks curl::parse_date()
```

```
## x jsonlite::toJSON()   masks rjson::toJSON(), RJSONIO::toJSON()
## x tidyr::unpack()      masks Matrix::unpack()

library(purrr)
library(tidyr)
library(dplyr)
library(reshape2)

## Warning: package 'reshape2' was built under R version 4.0.3

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

library(RColorBrewer)

## Warning: package 'RColorBrewer' was built under R version 4.0.3

part1 = data.frame(as.list(fromJSON(txt = "https://rafael.laboissiere.net/m1-
miashs-2021/Wax9le/data/2021-05-12-07-32-08.json")))
part2 = data.frame(as.list(fromJSON(txt ="https://rafael.laboissiere.net/m1-
miashs-2021/Wax9le/data/2021-05-13-19-27-50.json" )))
part3 = data.frame(as.list(fromJSON(txt = "https://rafael.laboissiere.net/m1-
miashs-2021/Wax9le/data/2021-05-13-20-10-26.json")))
part4 = data.frame(as.list(fromJSON(txt = "https://rafael.laboissiere.net/m1-
miashs-2021/Wax9le/data/2021-05-13-22-24-49.json")))
part5 = data.frame(as.list(fromJSON(txt = "https://rafael.laboissiere.net/m1-
miashs-2021/Wax9le/data/2021-05-16-15-14-26.json")))
part6 = data.frame(as.list(fromJSON(txt = "https://rafael.laboissiere.net/m1-
miashs-2021/Wax9le/data/2021-05-16-19-23-07.json")))




#met tous les data frames des participants dans un seul dataframe
tousPart = rbind(part1, part2, part3, part4, part5, part6)
##

##creation d'un data frame

subject <- c(tousPart$id)
frequency <- c(2)
congruency <- c(tousPart$essais.congruent)
MT <- c(tousPart$essais.reactionTime)
AUC <- c()
erreurs <- c(tousPart$essais.erreur)
IT <- c(tousPart$essais.tempsInit)
groupe <- c(tousPart$groupe)
```

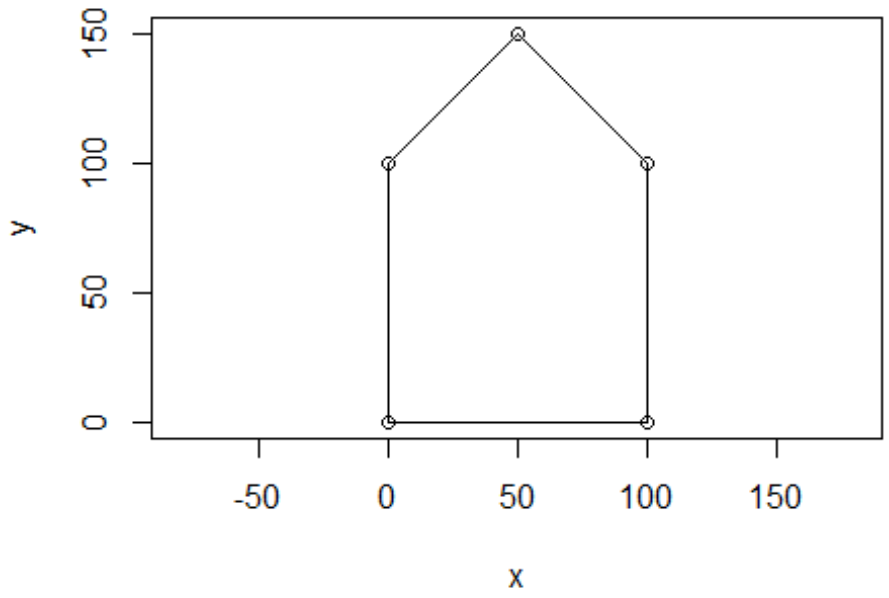
```r
############################################################
### fonction pour calculer l'aire sous la courbe

polyarea <- function (x, y) {
  n <- length (x)
  return (abs (sum (x [1 : (n - 1)] * y [2 : n] - y [1 : (n - 1)] * x [2 :
n])) / 2)
}
x <- c (0, 0, 50, 100, 100, 0)
y <- c (0, 100, 150, 100, 0, 0)
plot (x, y, asp = 1)
lines (x, y)
```




```r
polyarea (x, y)
```

```
## [1] 12500
```

```r
x <- c (0, 0, 50, 100, 100, 0)
y <- c (0, 100, 150, 100, 0, 0)
plot (x, y, asp = 1)
lines (x, y)
```

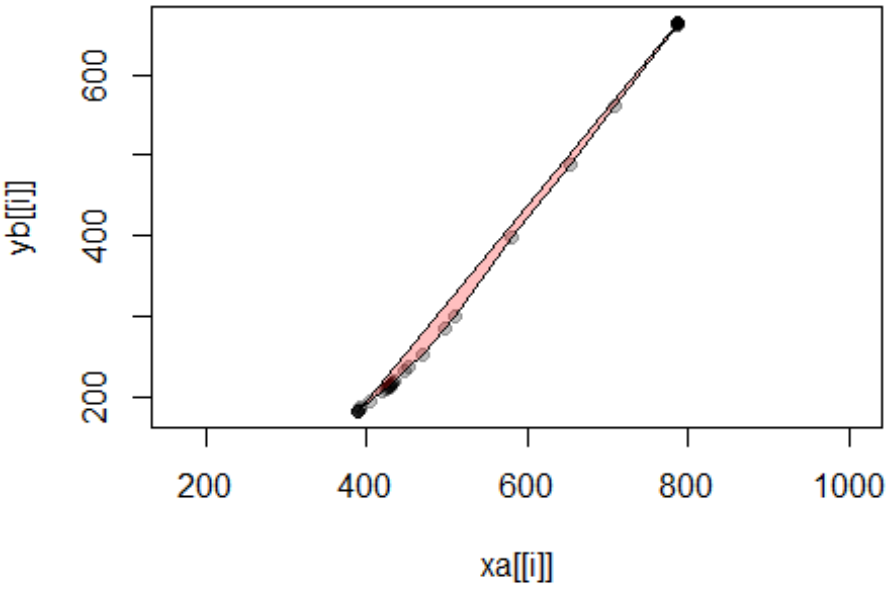


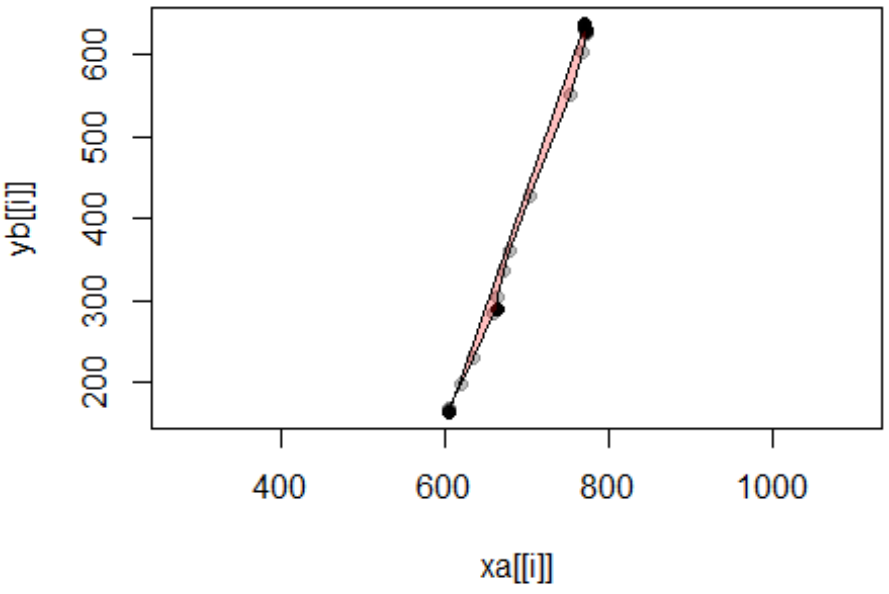

```r
polyarea (x, y)
```

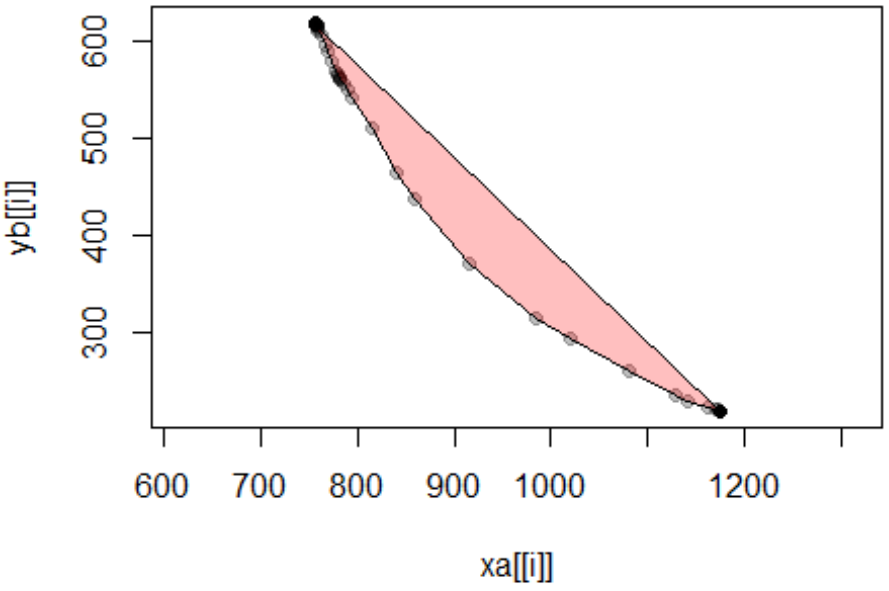
```
## [1] 12500
```

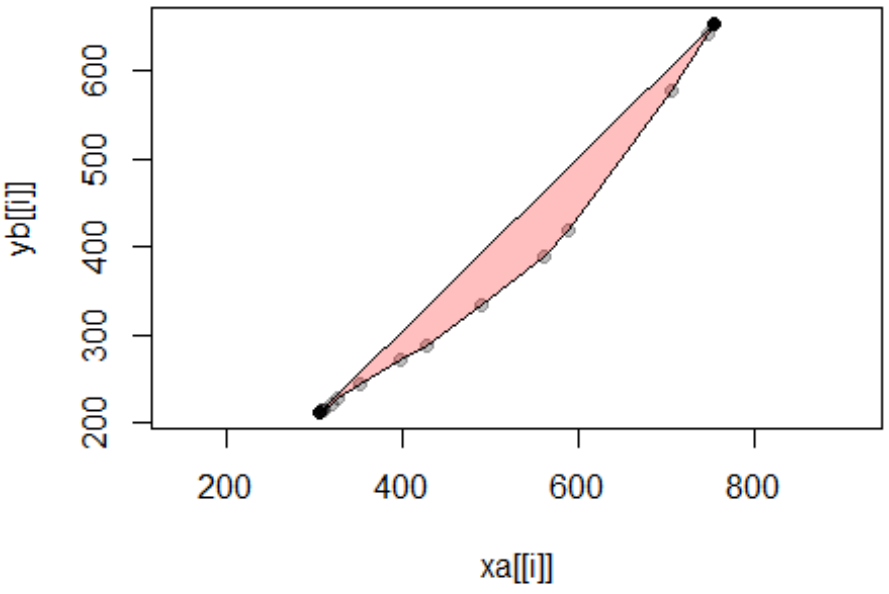
```r
#################################################################


#################################################################
#boucle qui affiches tous les courbes et calcule l'aires sous la courbe de
chaque essais

xa <- c()
yb <- c()
loop <- tousPart

for(i in 1:nrow(loop)) {       # for-loop over rows

  xa[i] <- loop[i, 8]
  yb[i] <- loop[i, 9]

  plot (xa[[i]], yb[[i]], pch = 19, col = "#00000040", asp = 1)

  xa[[i]] <- c (xa[[i]], xa[[i]][1])
  yb[[i]] <- c (yb[[i]], yb[[i]][1])

  lines (xa[[i]], yb[[i]])
```

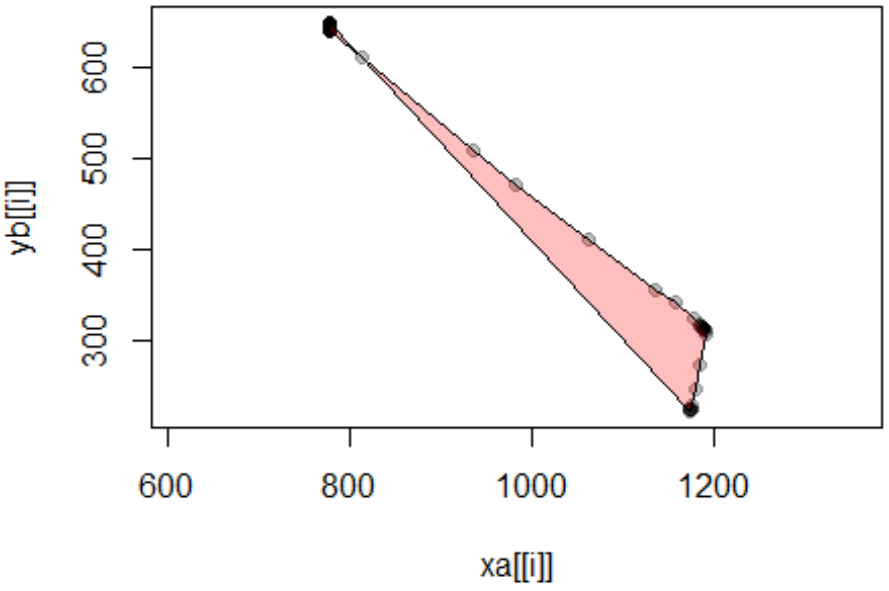
```
  AUC[i] <- polyarea (xa[[i]], yb[[i]])

  polygon (xa[[i]], yb[[i]], col = "#ff000040")


}
```

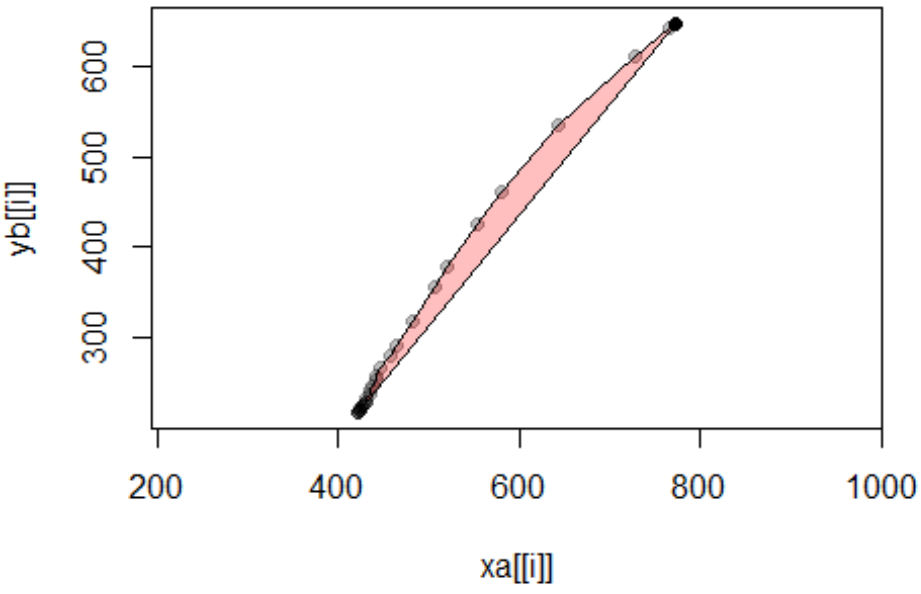

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

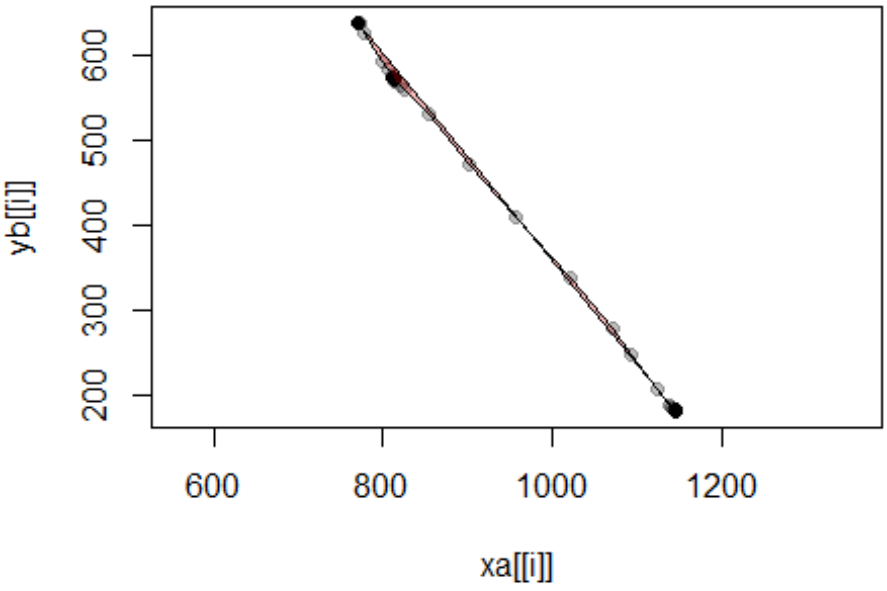

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

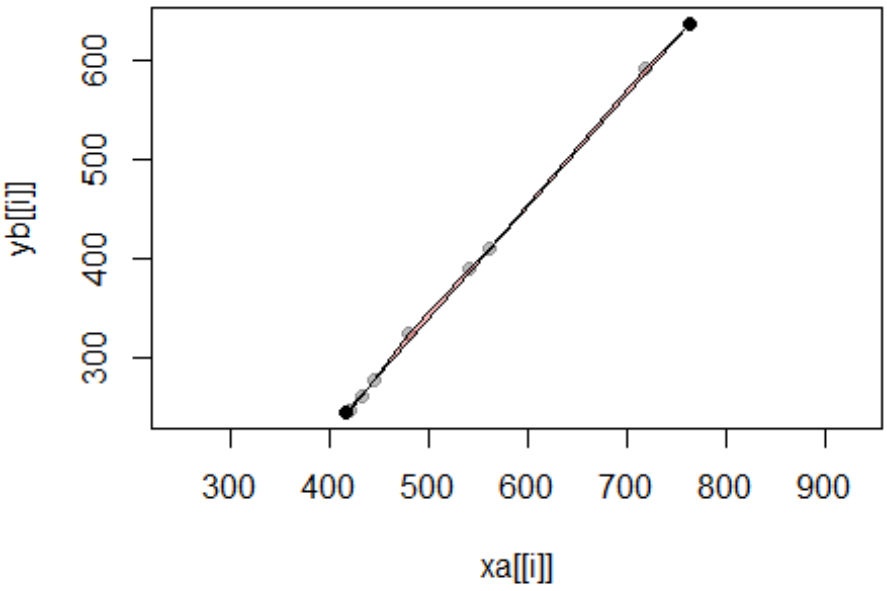

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

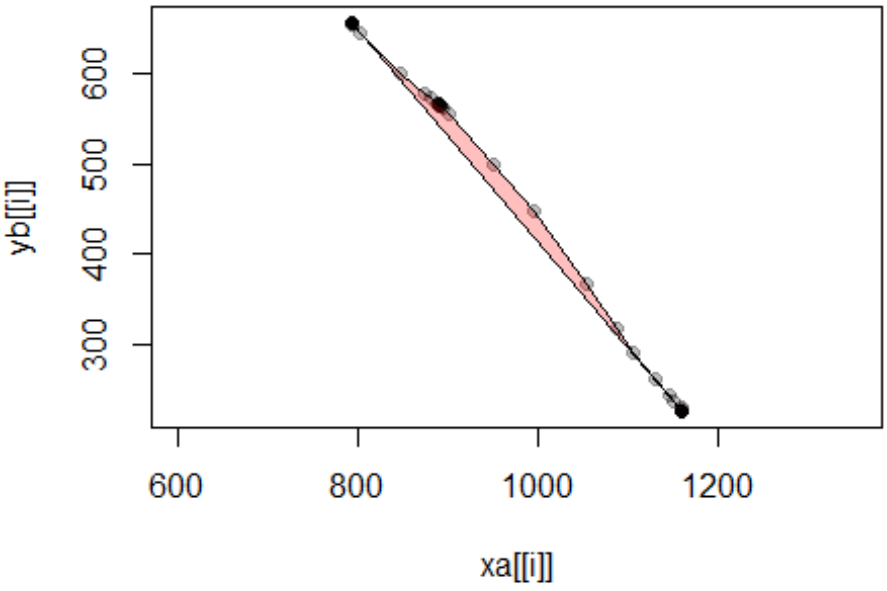

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

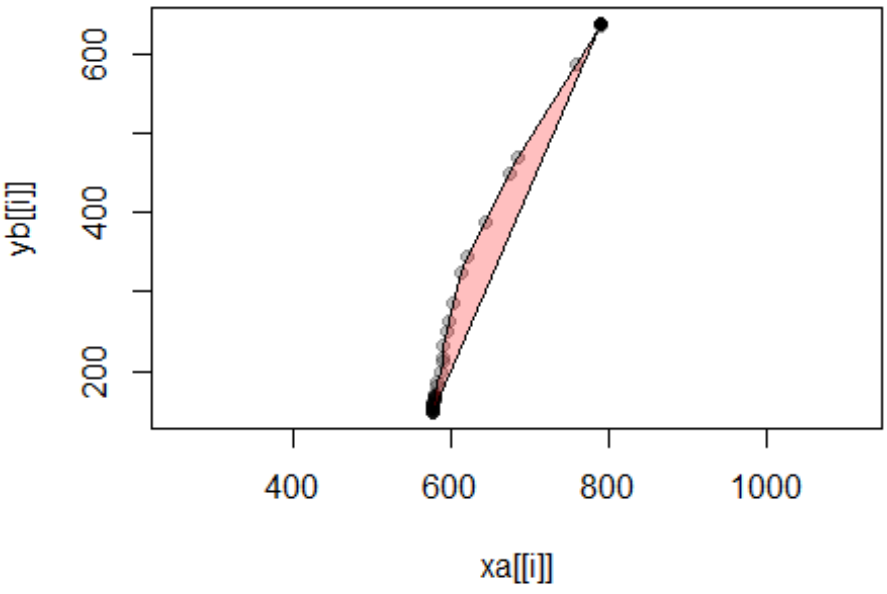

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

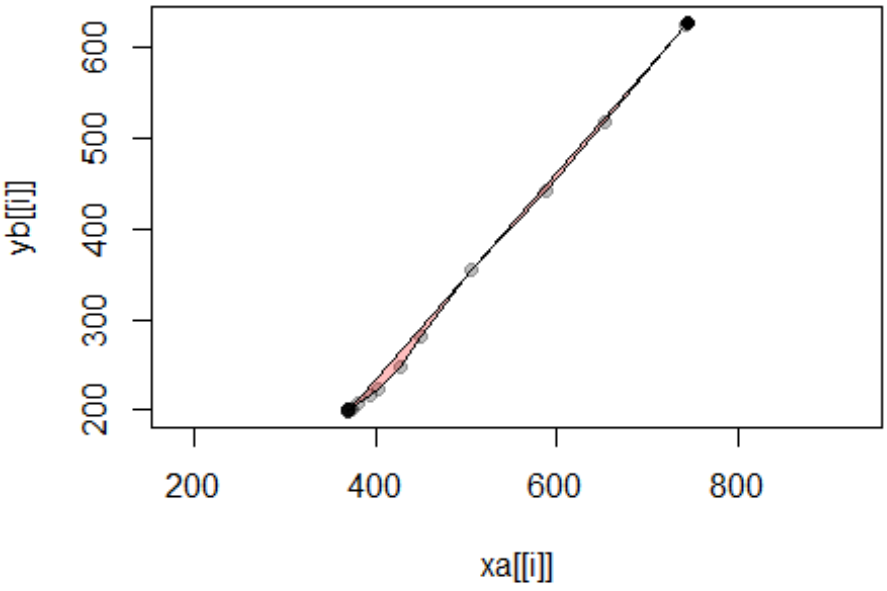

yb[[i]]
300
400
500
600
200
400
600
800
1000
xa[[i]]

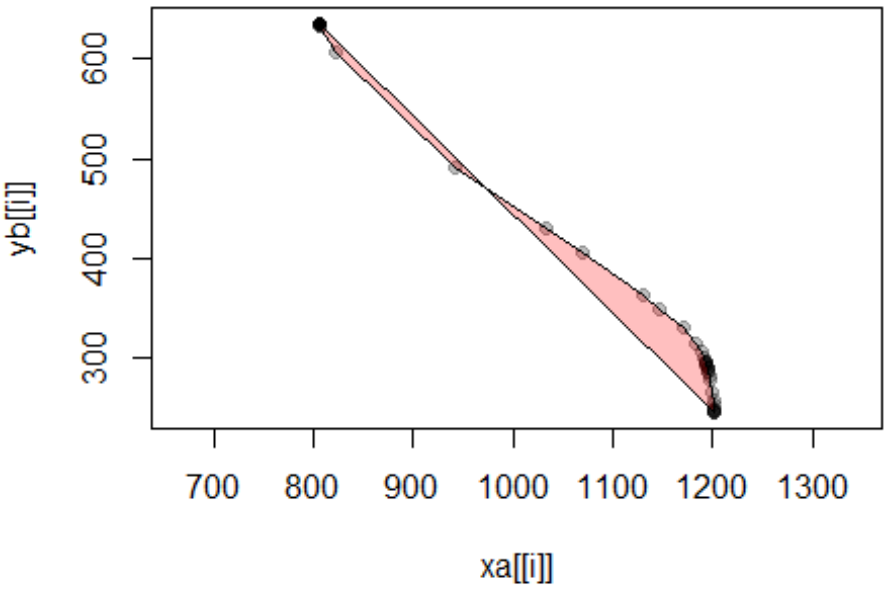

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

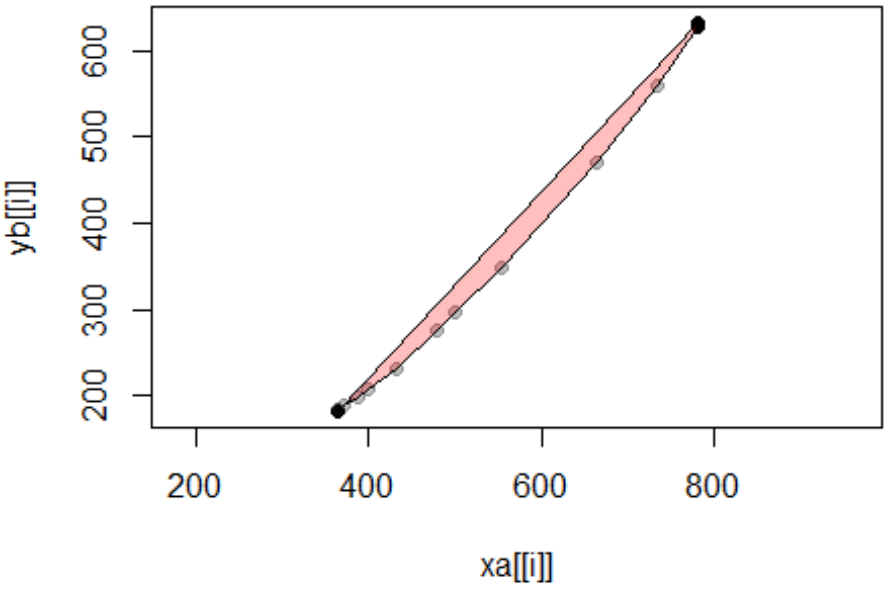

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

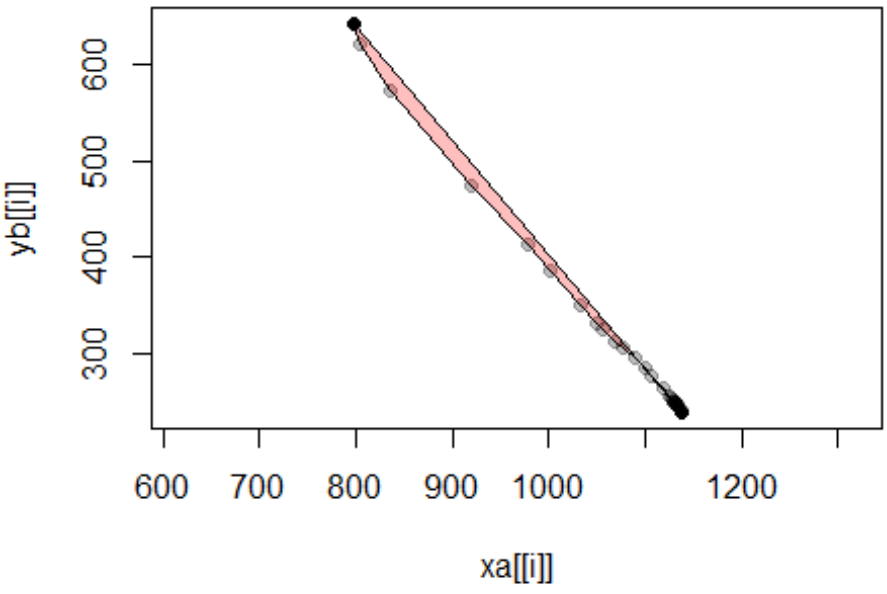

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

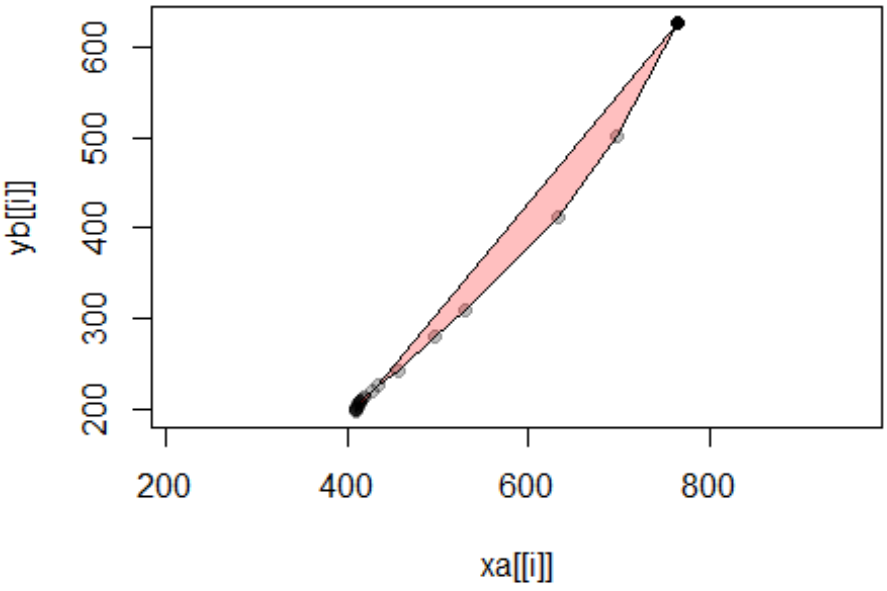

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

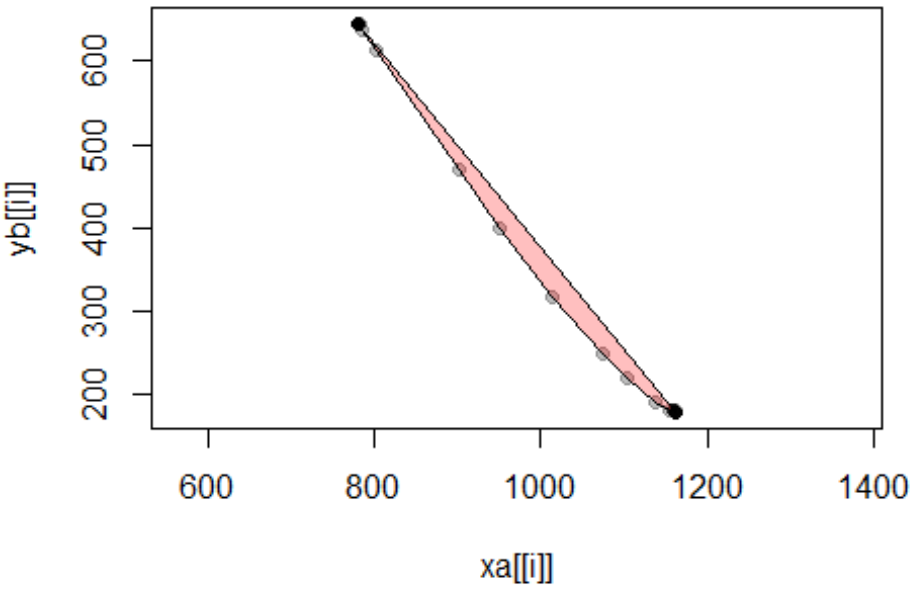

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

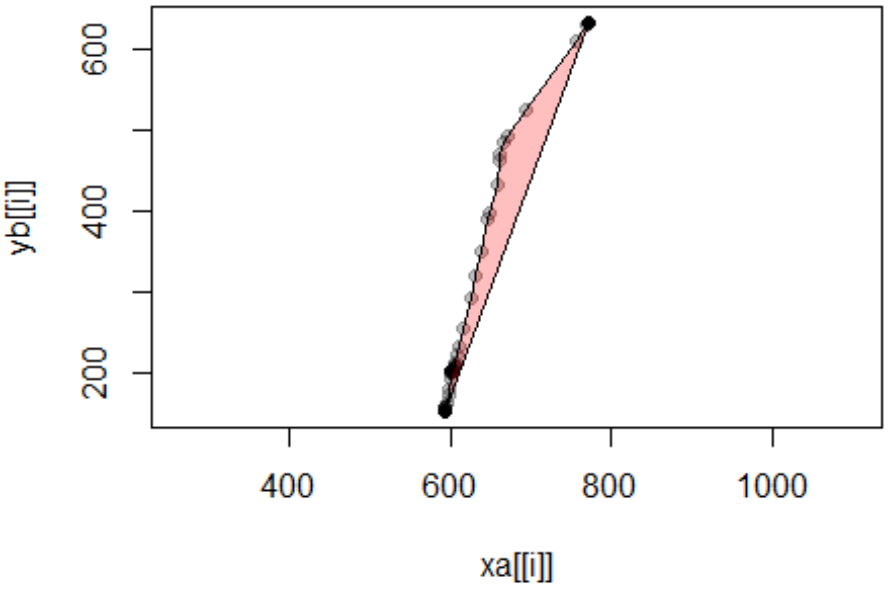

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

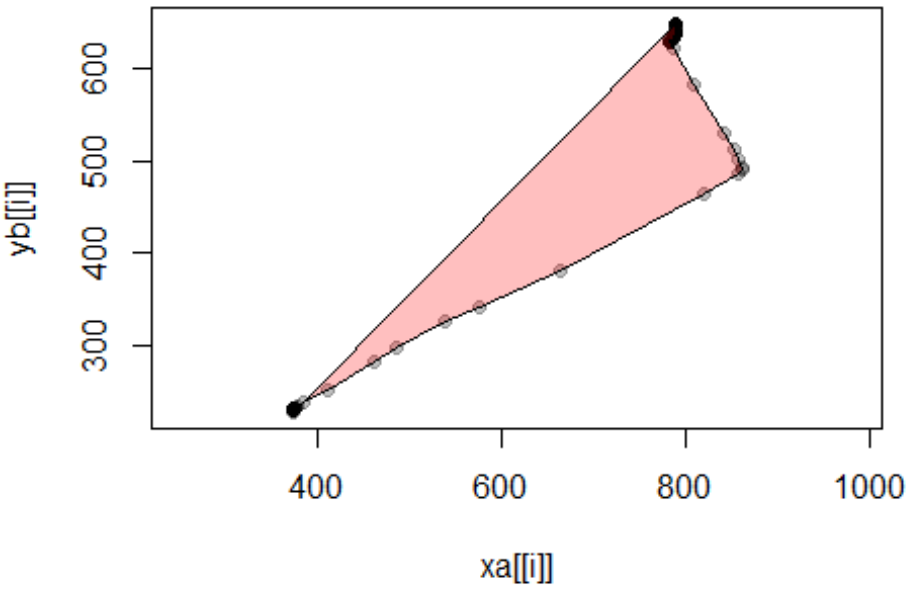

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

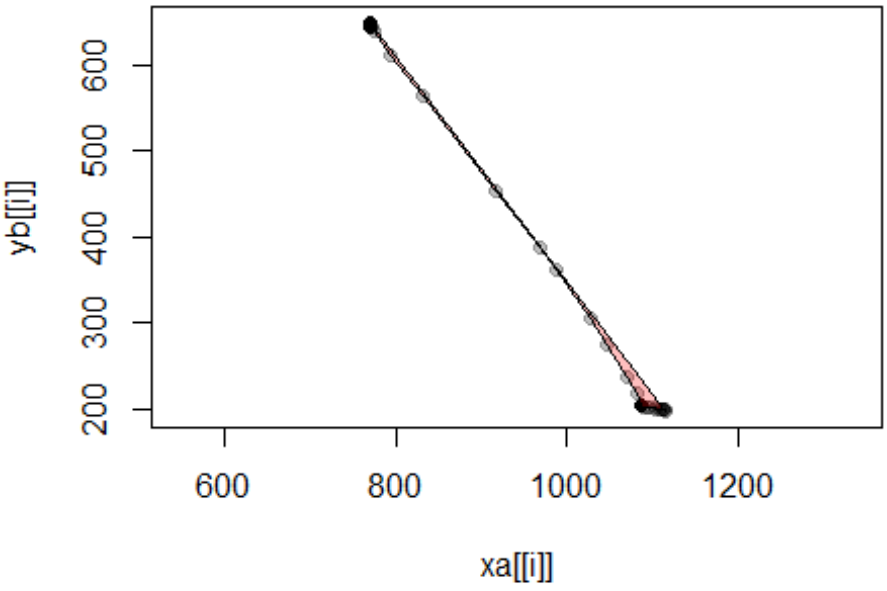

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

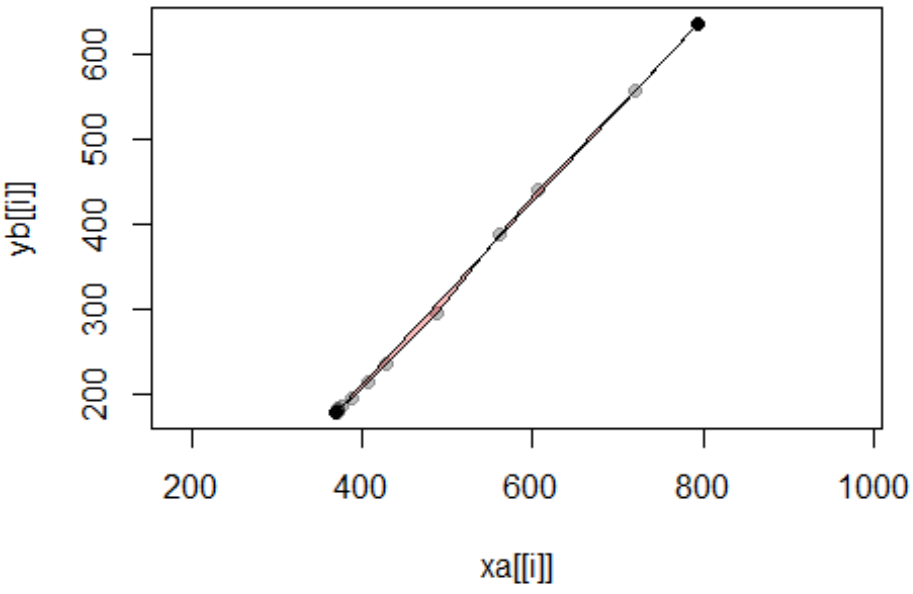

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

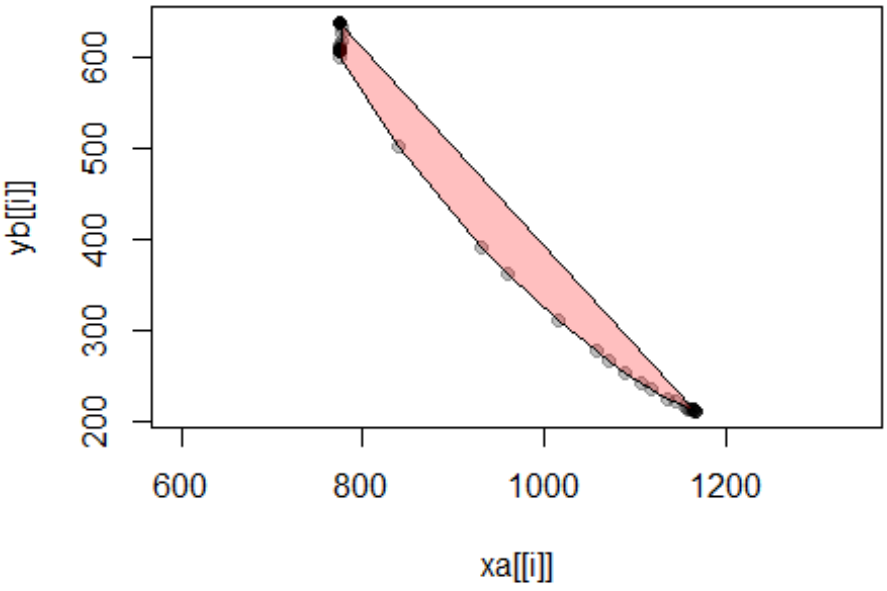

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

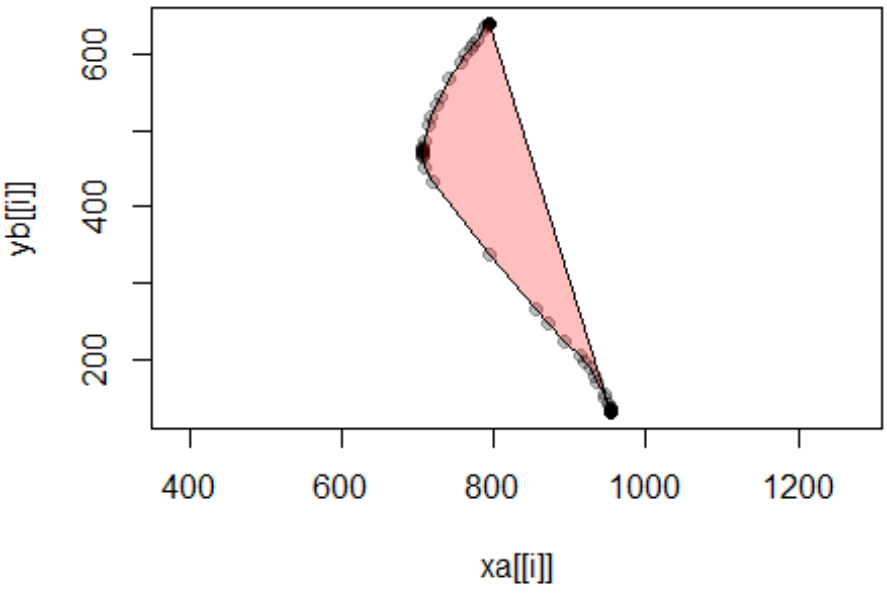

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

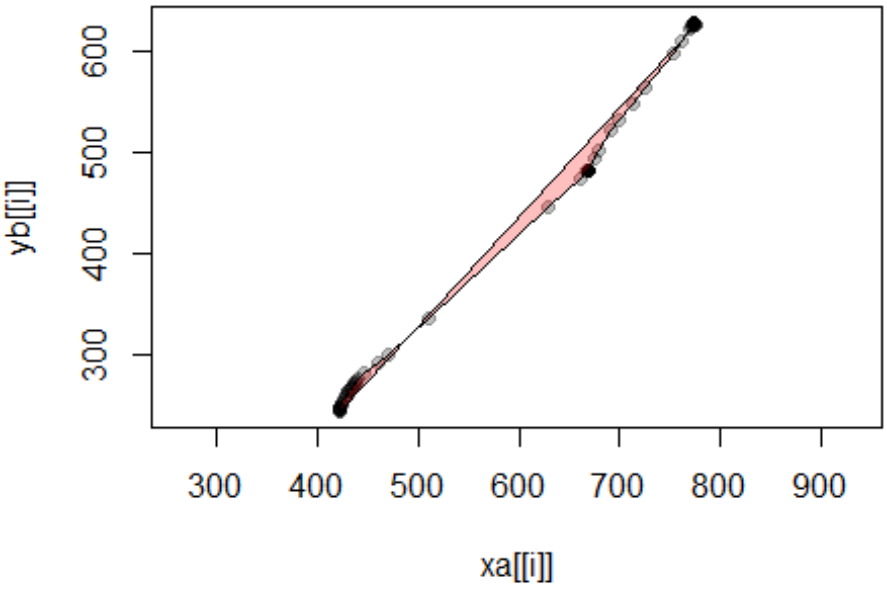

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

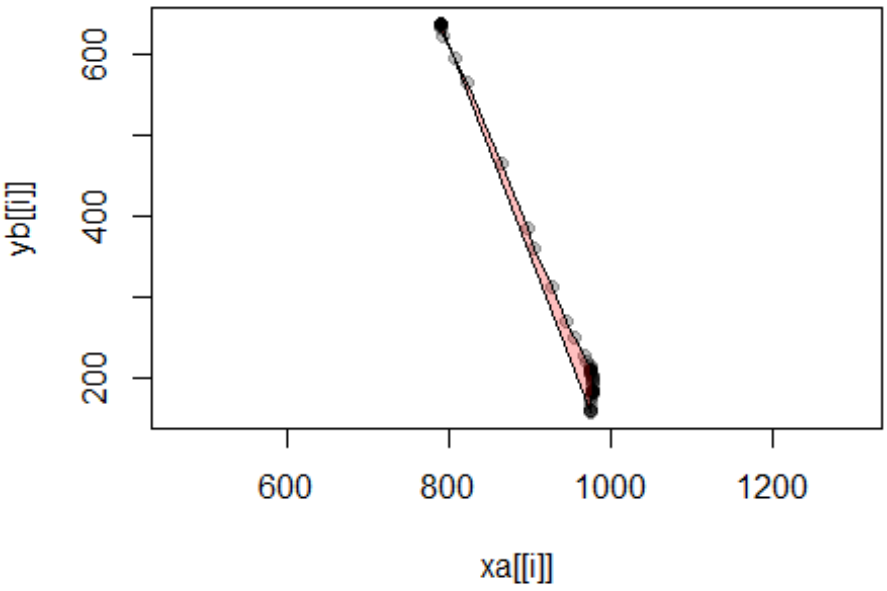

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

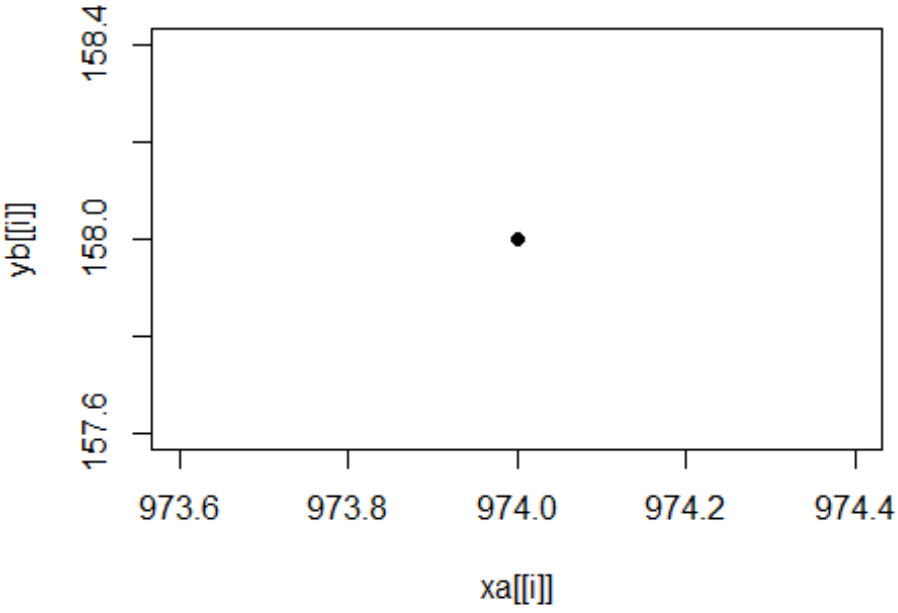

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

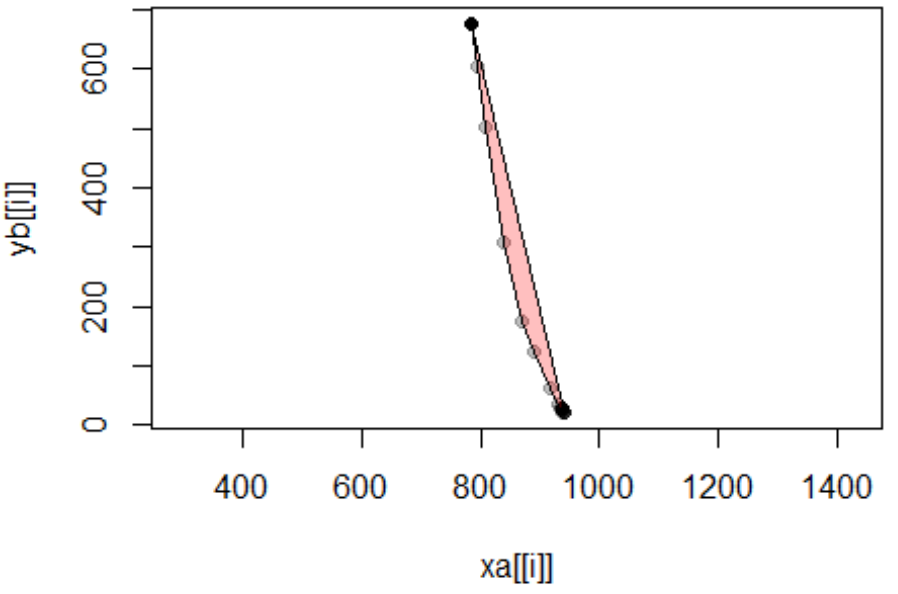

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

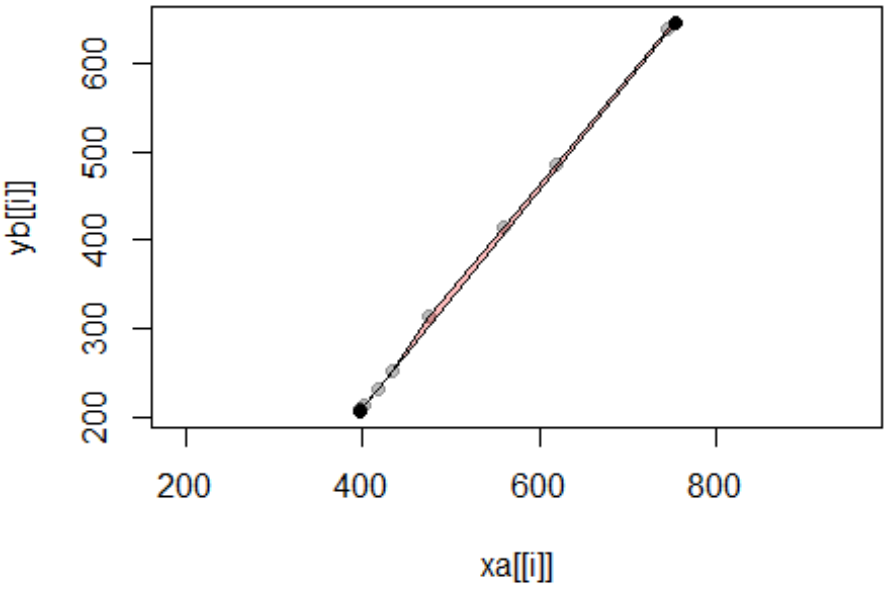

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

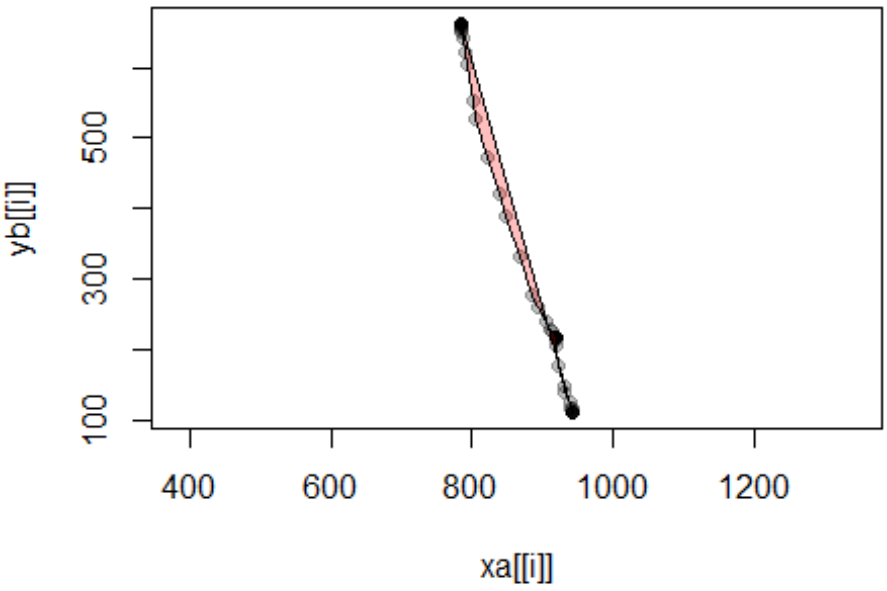

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

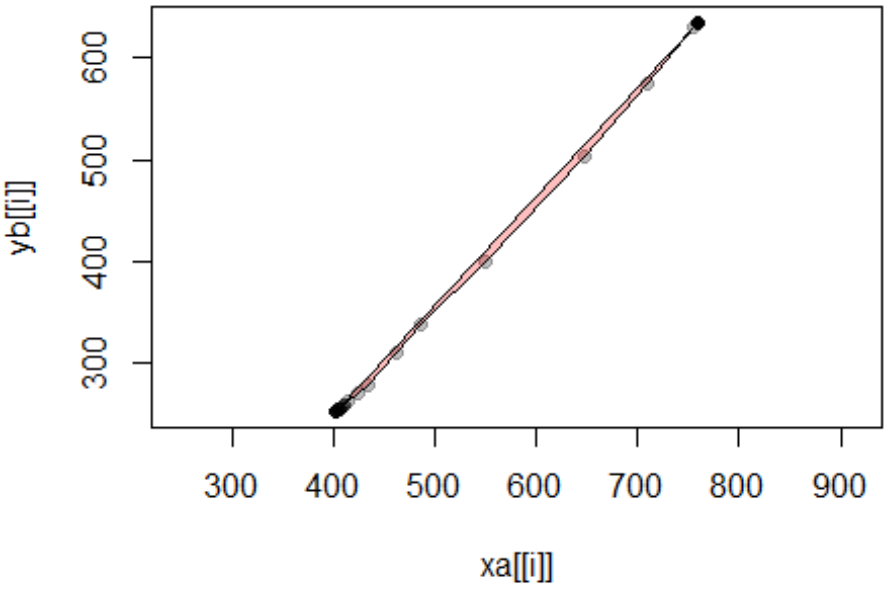

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

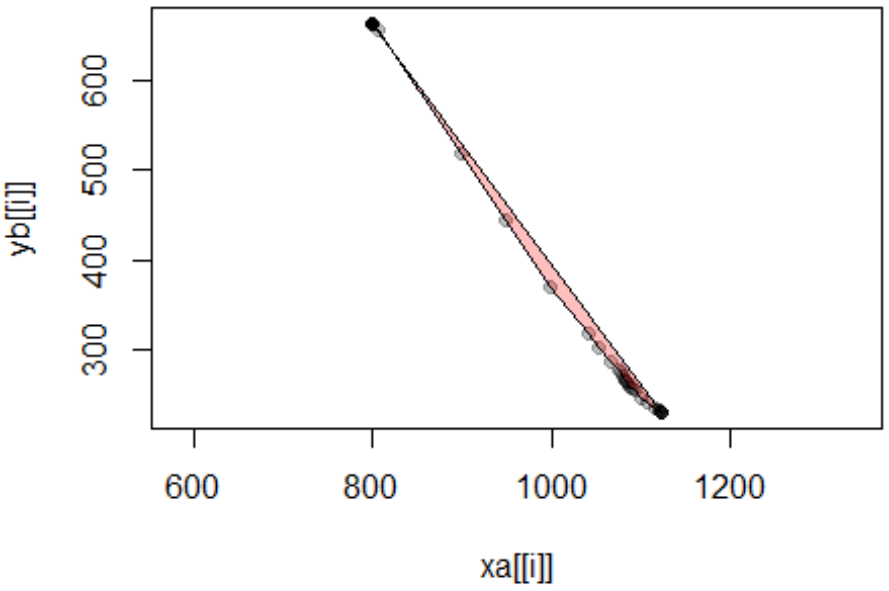

158.4
158.0
157.6
yb[[i]]
973.6
973.8
974.0
974.2
974.4
xa[[i]]

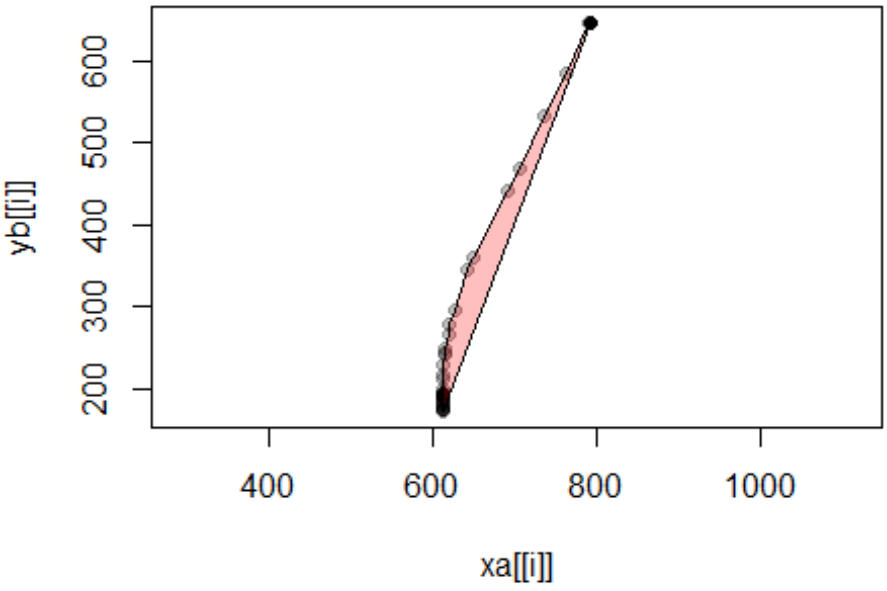

600
400
200
0
yb[[i]]
400
600
800
1000
1200
1400
xa[[i]]

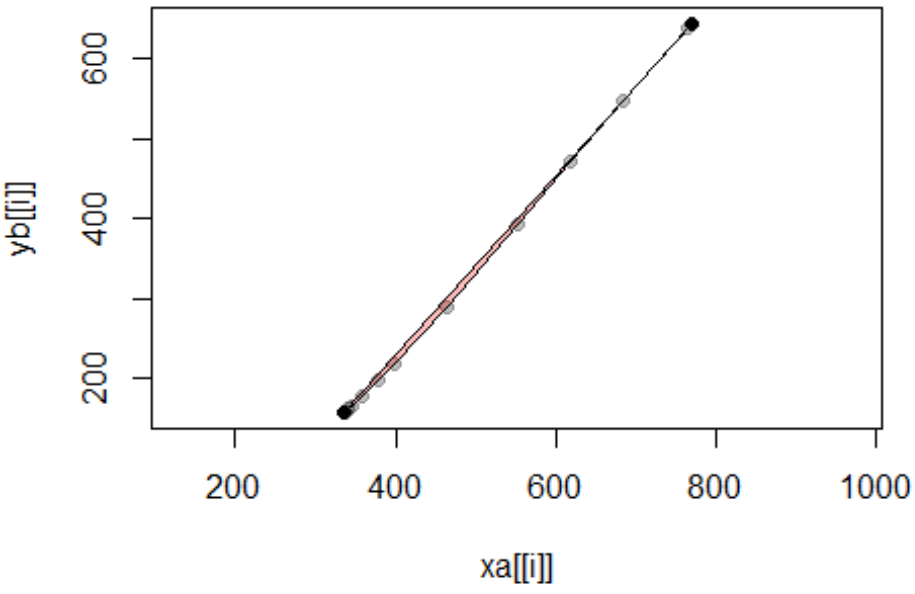

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

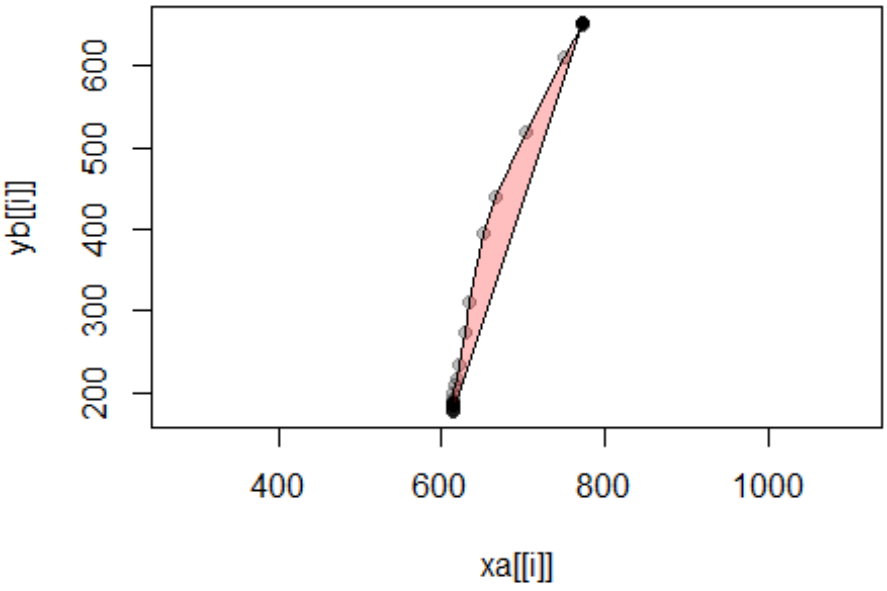

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

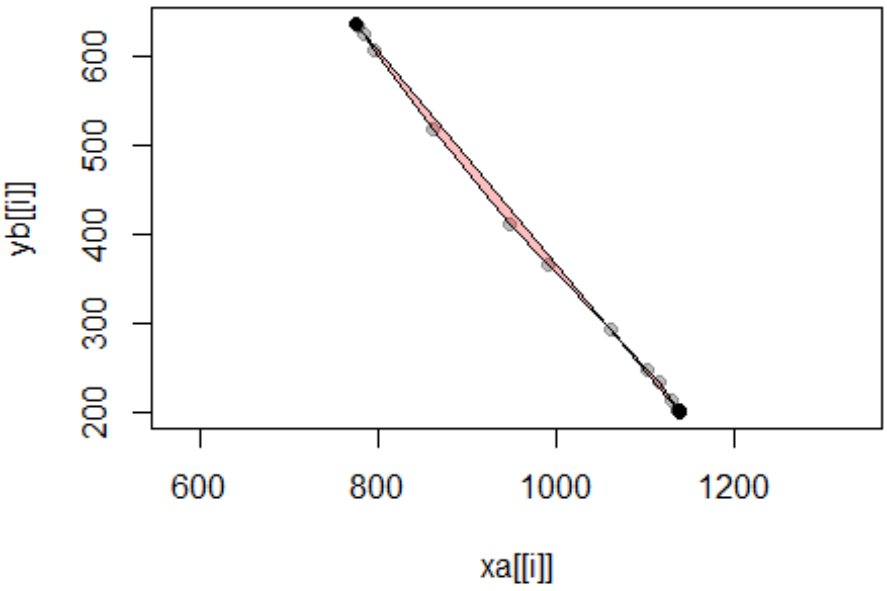

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

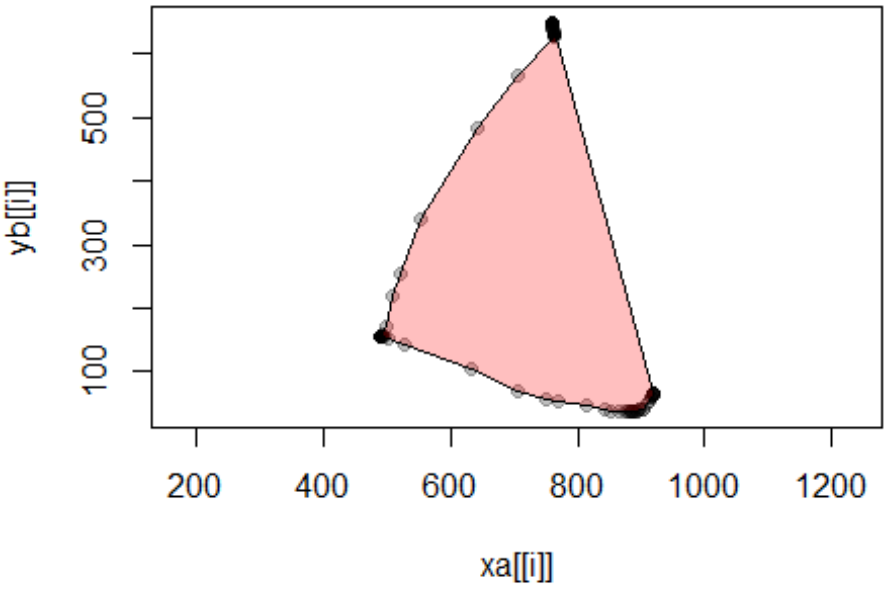

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

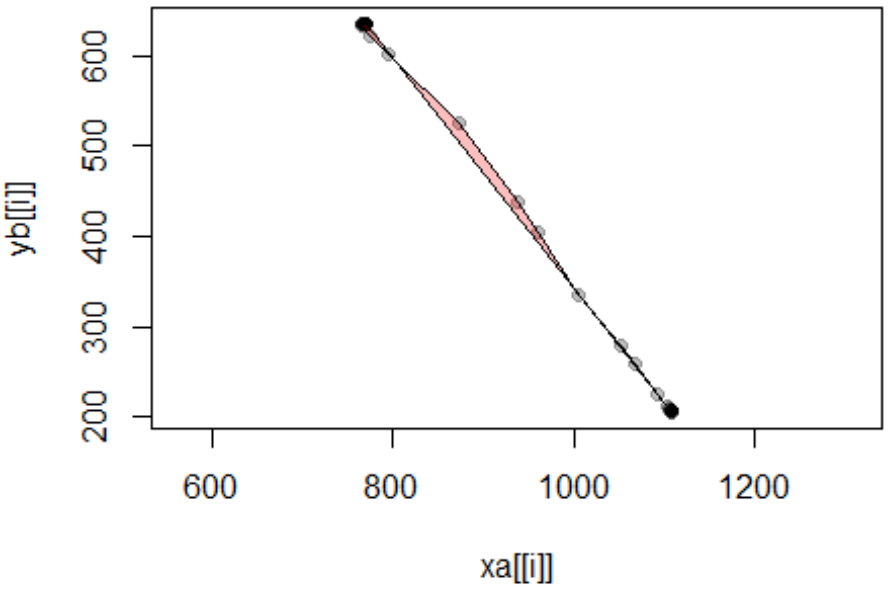

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

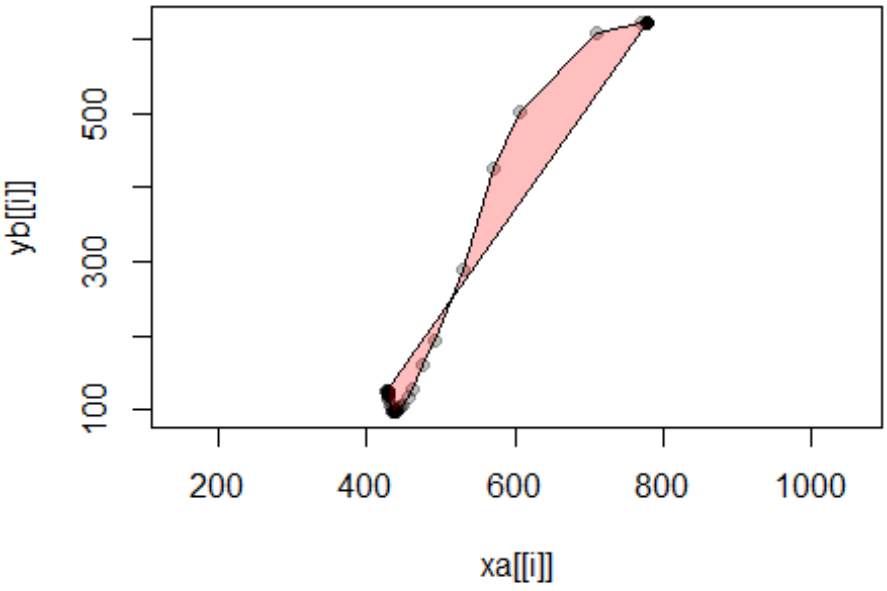

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

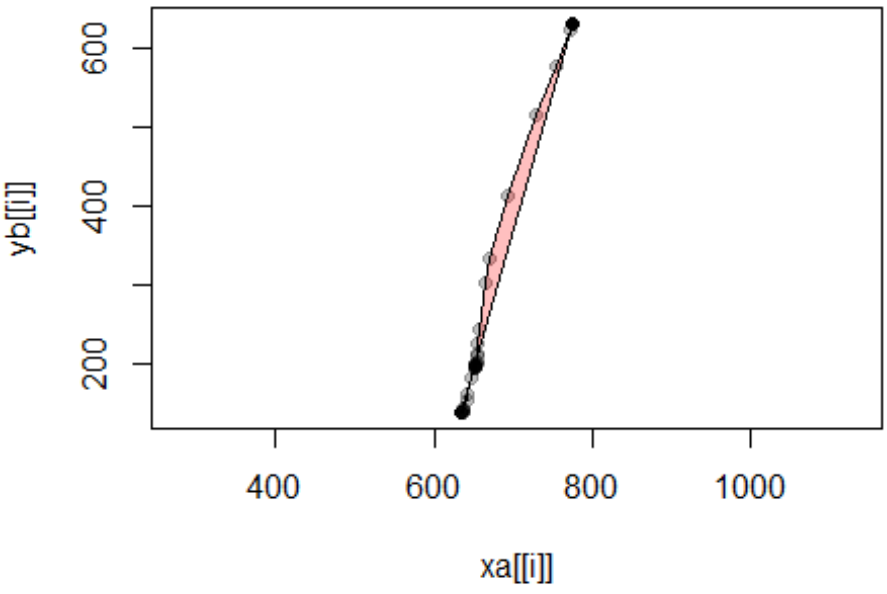

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

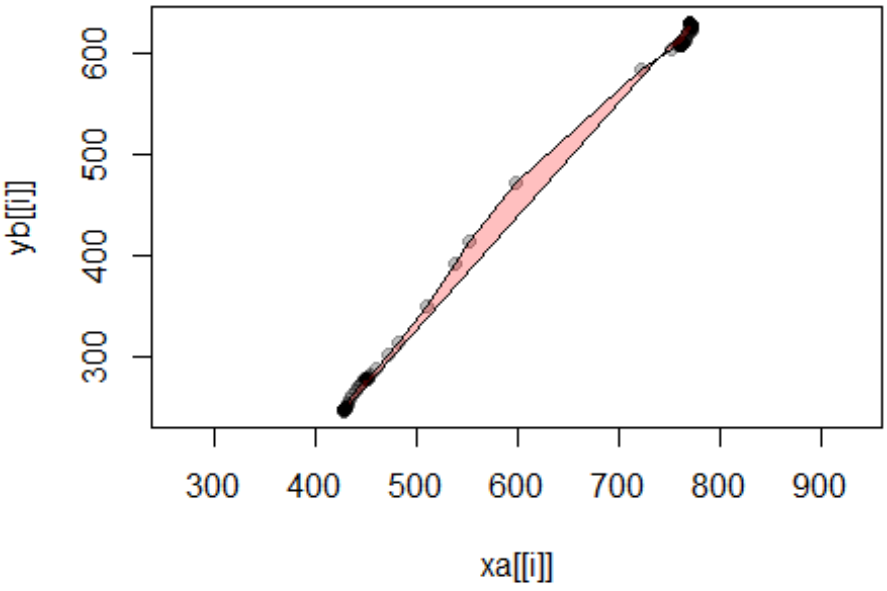

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

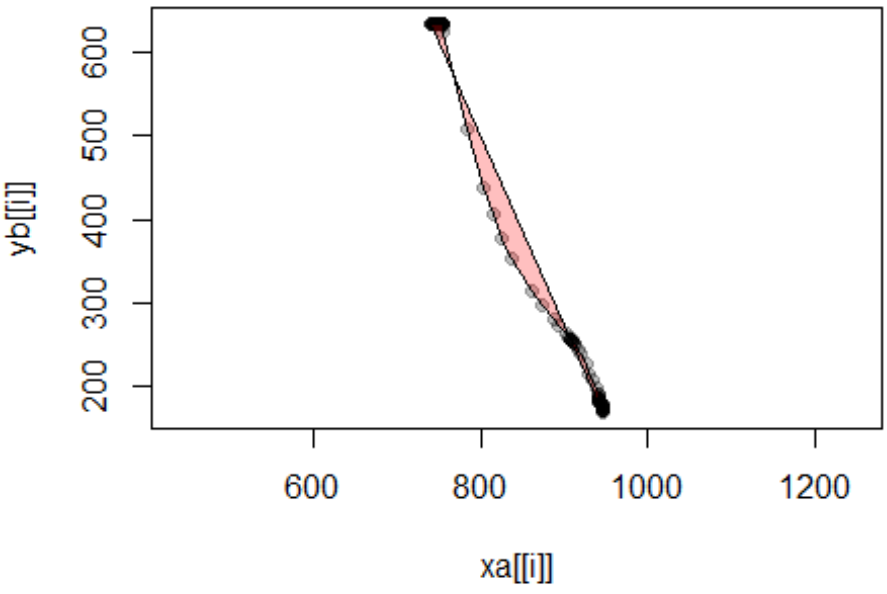

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

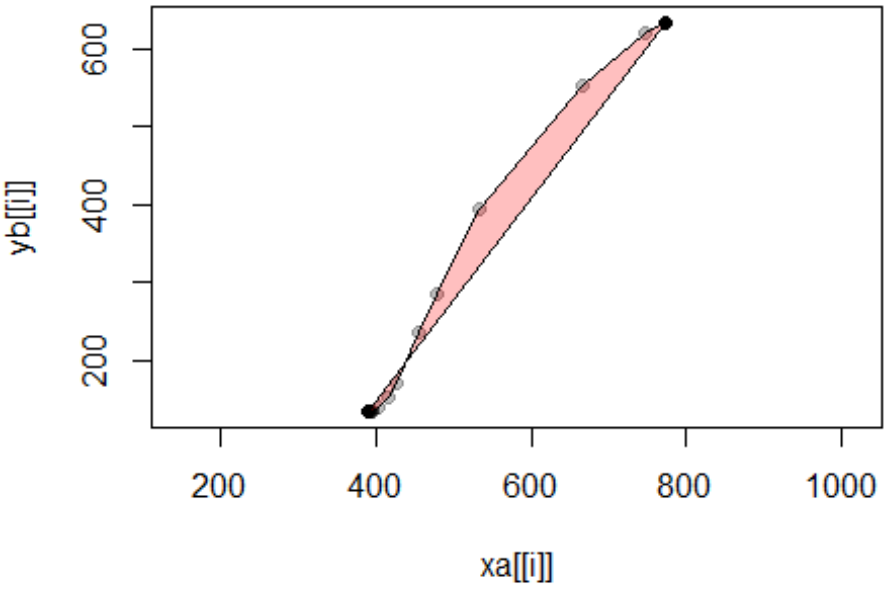

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

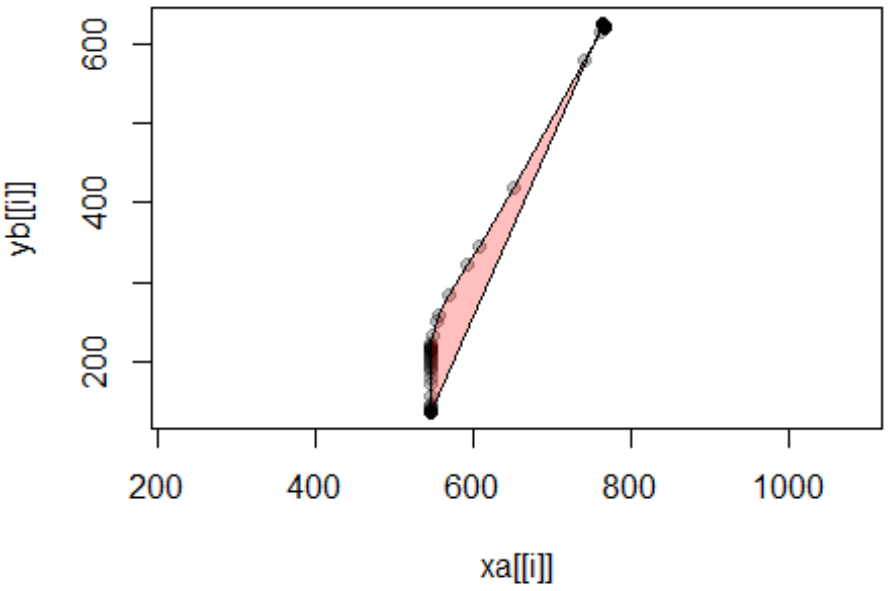

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

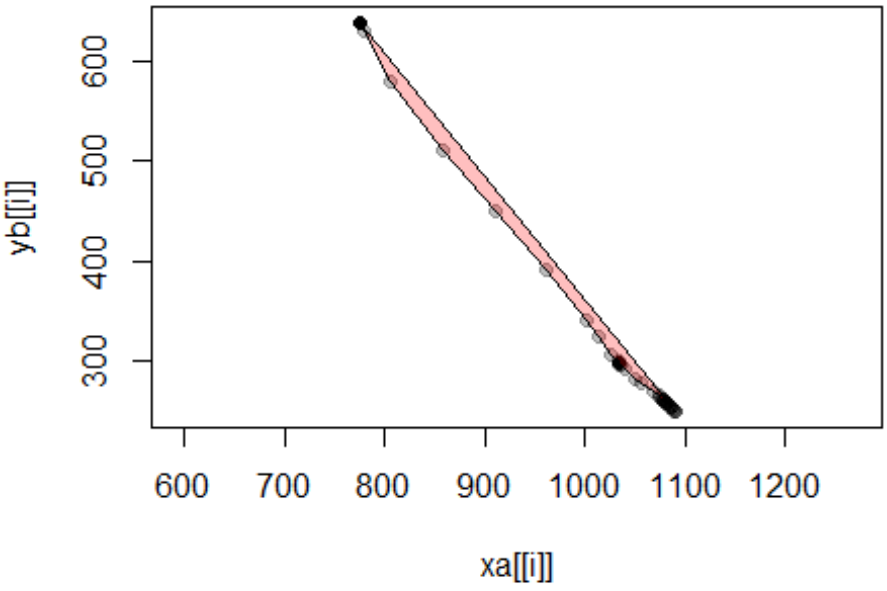

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

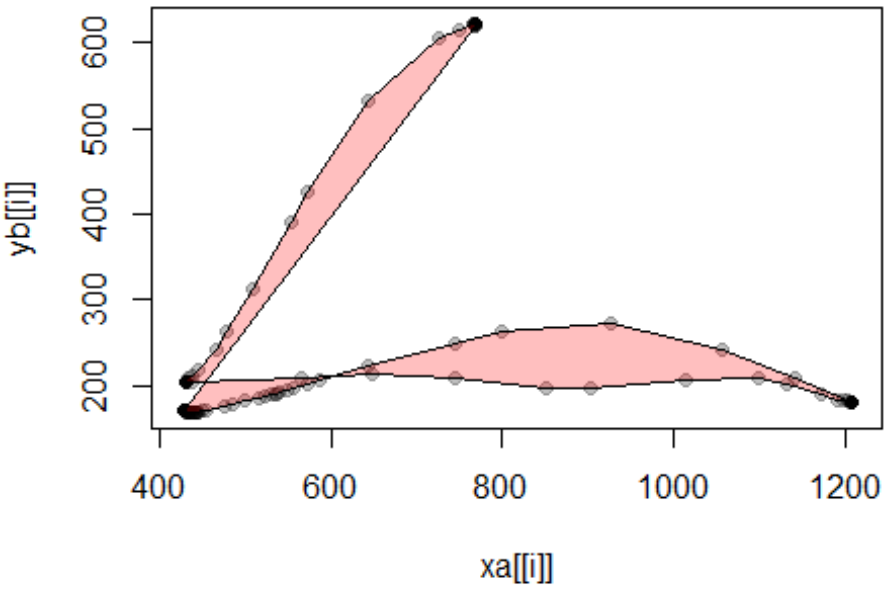

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

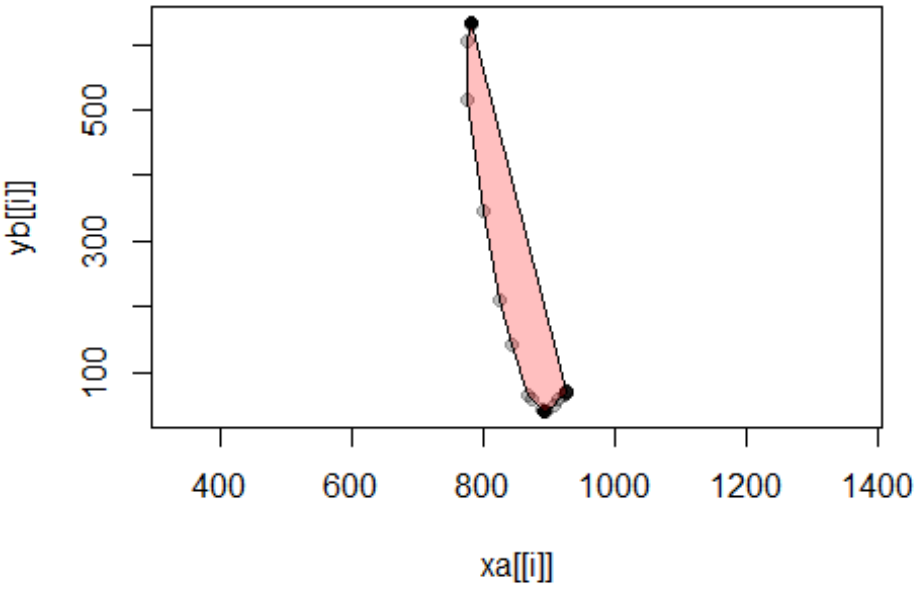

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

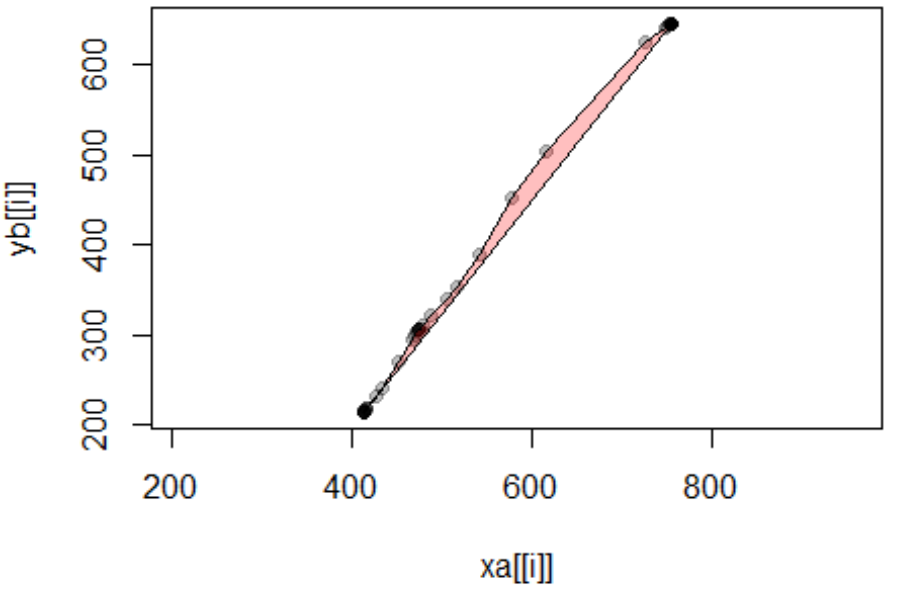

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

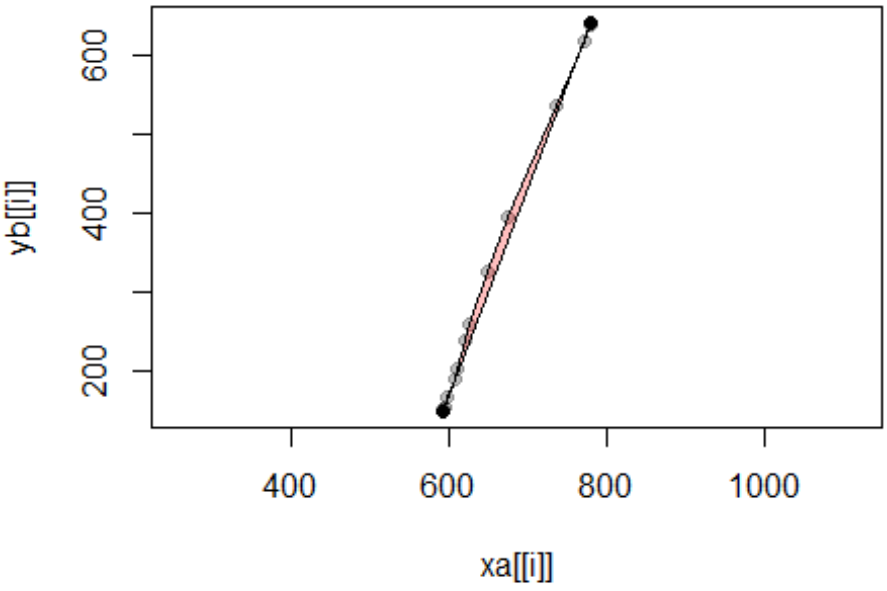

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

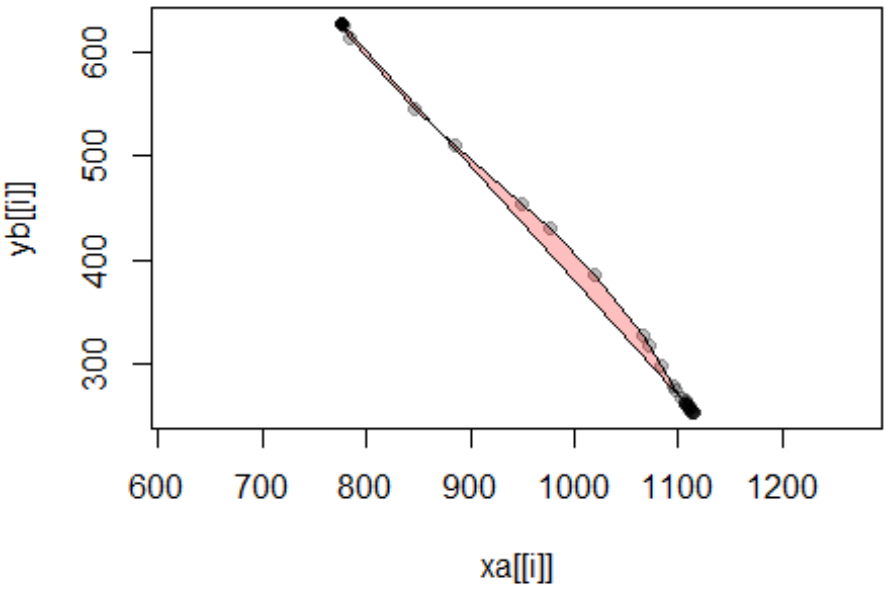

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
xa[[i]]

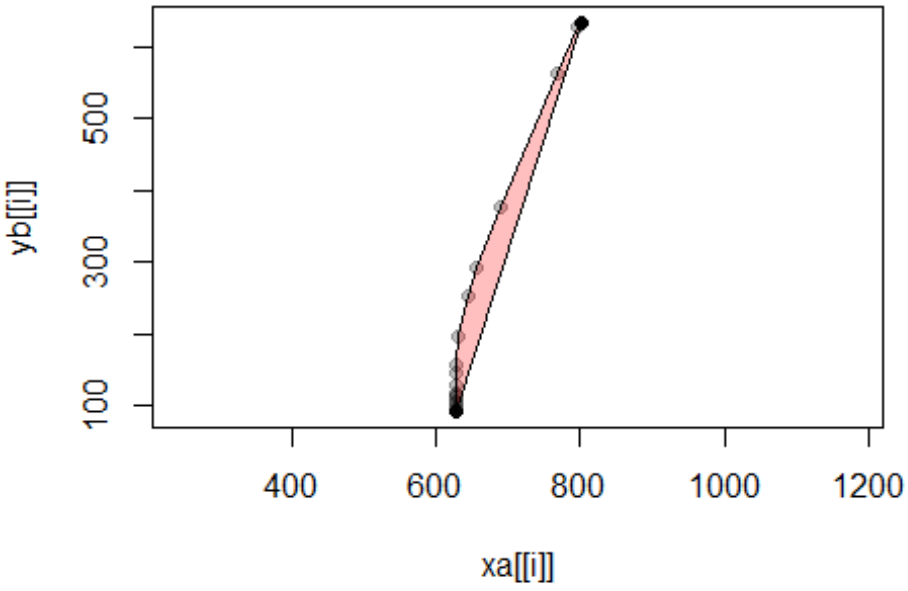

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

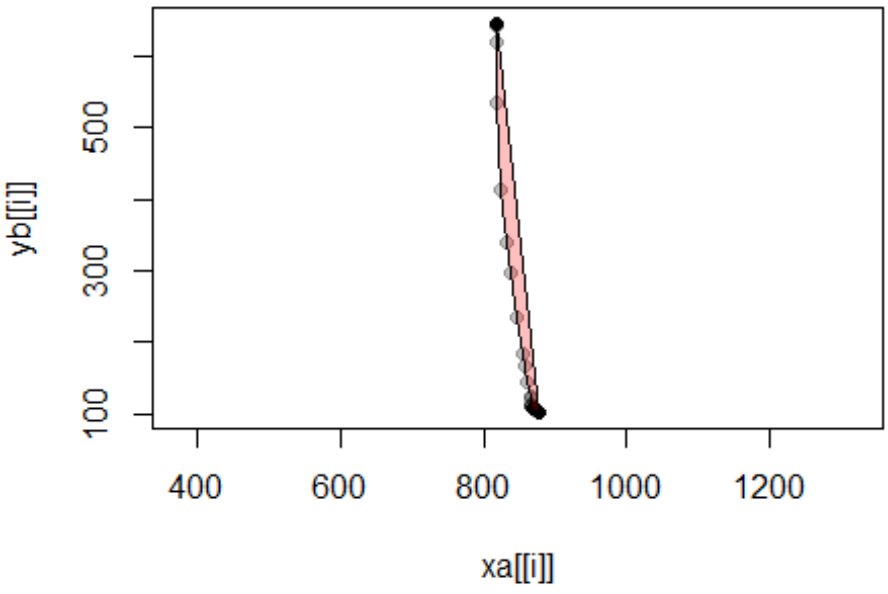

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

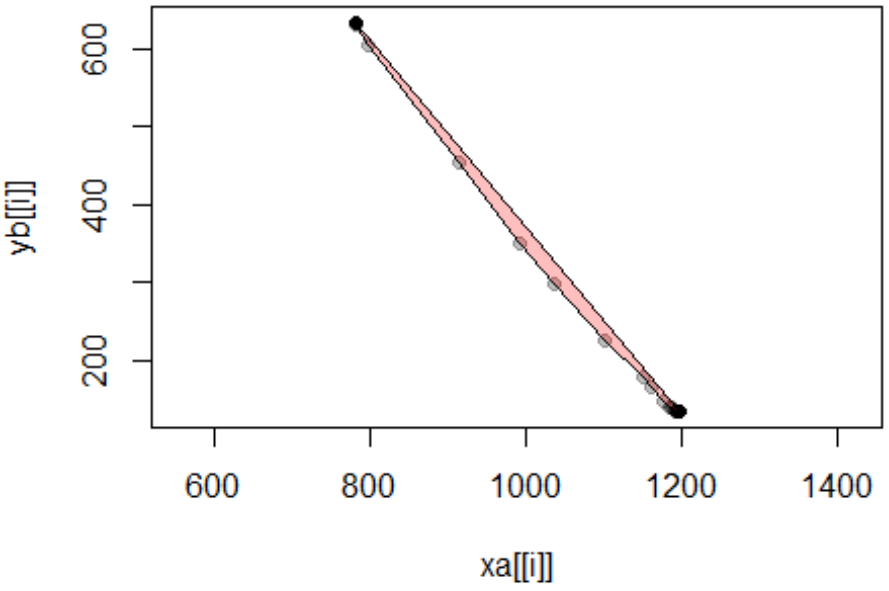

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

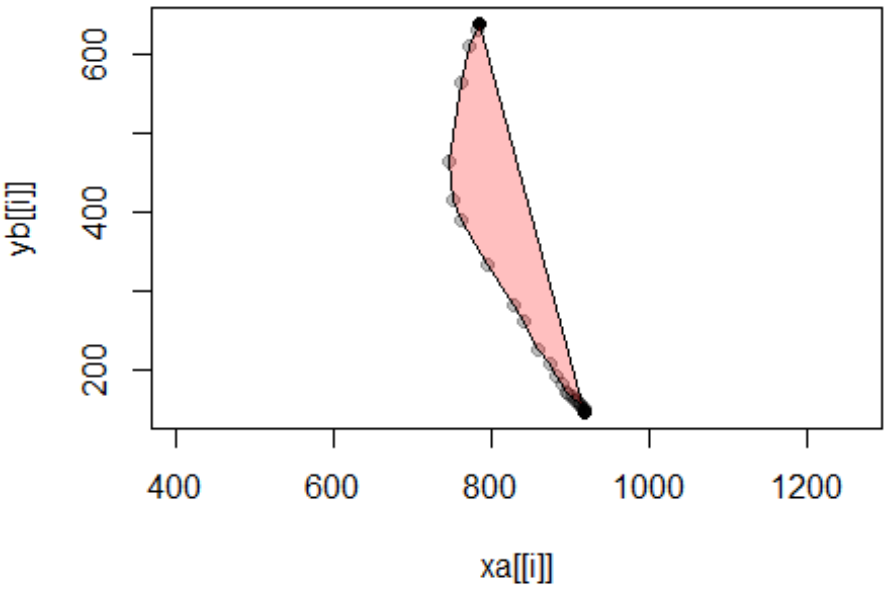

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

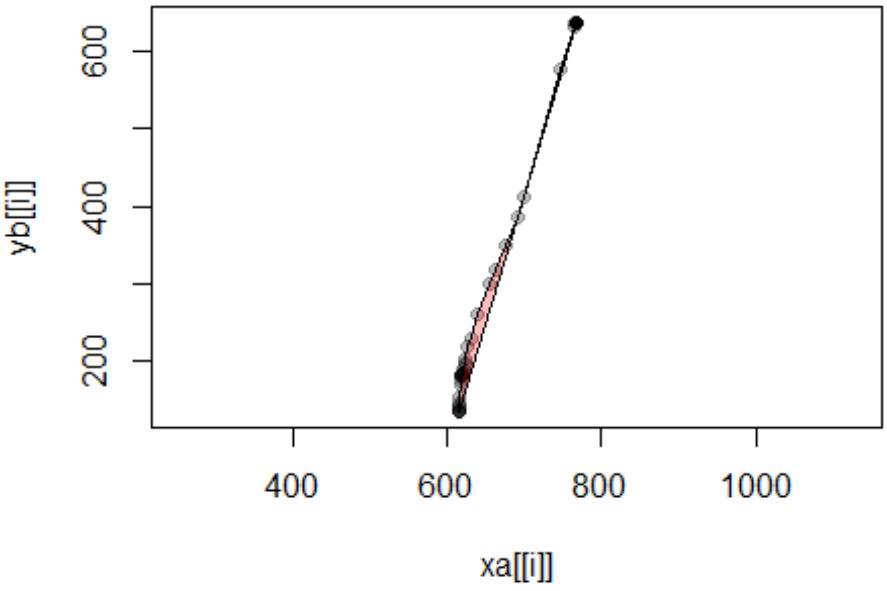

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
xa[[i]]

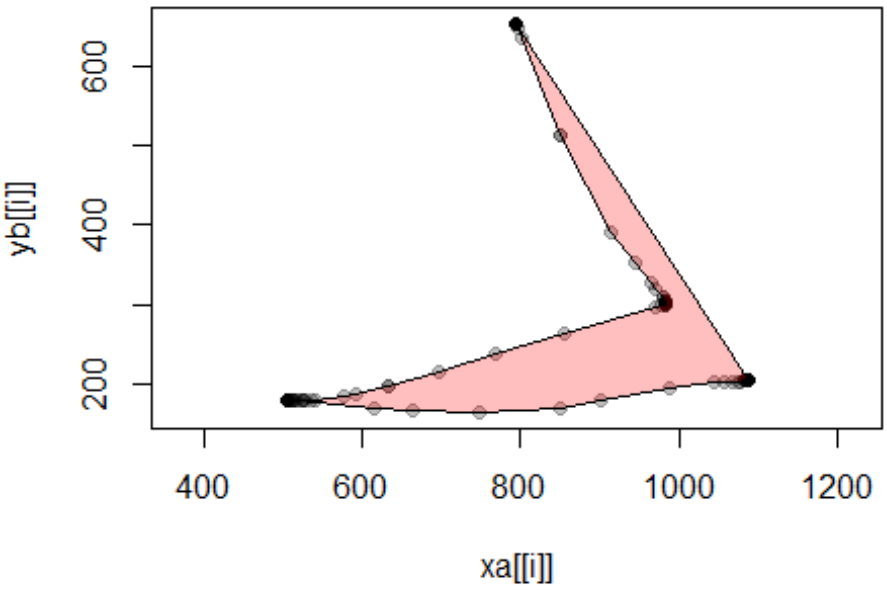

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

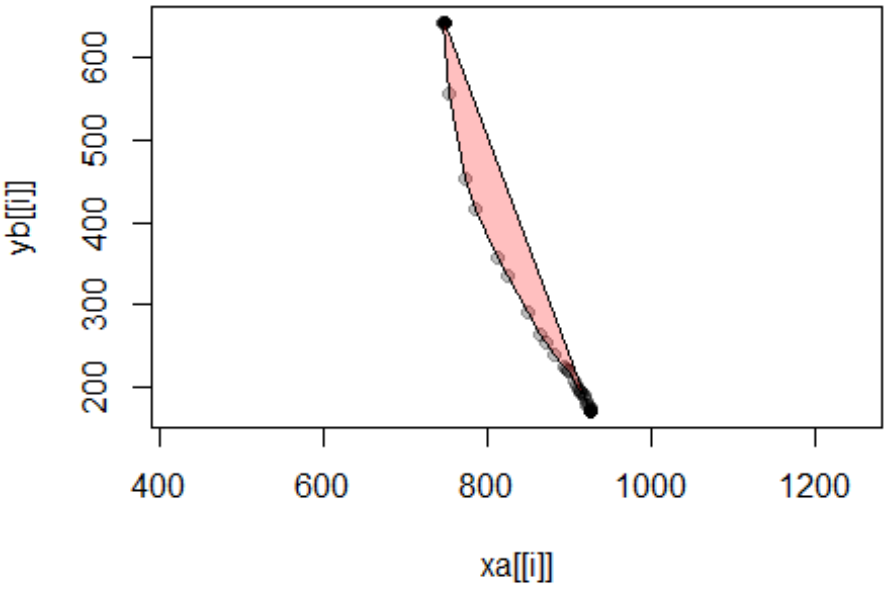

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

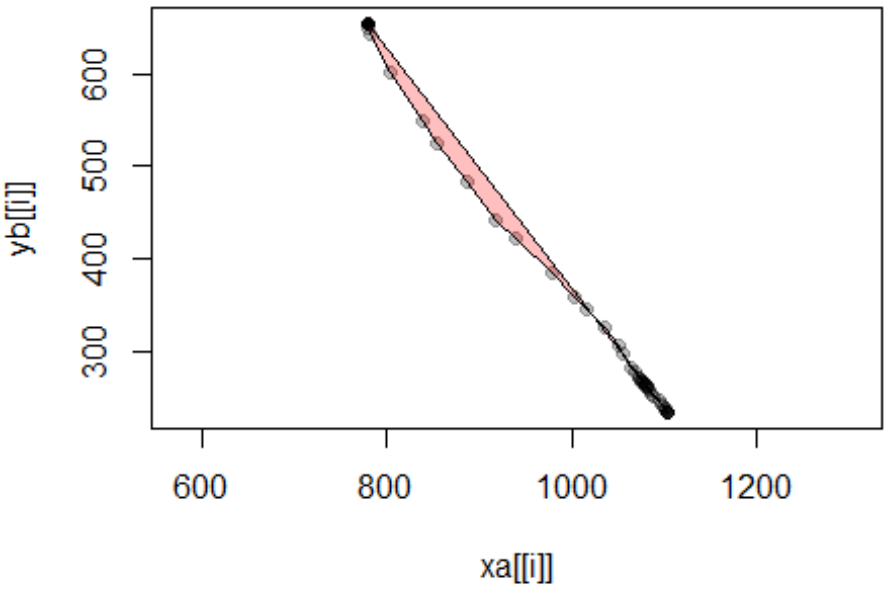

yb[[i]]
600
400
200
600
800
1000
1200
1400
xa[[i]]

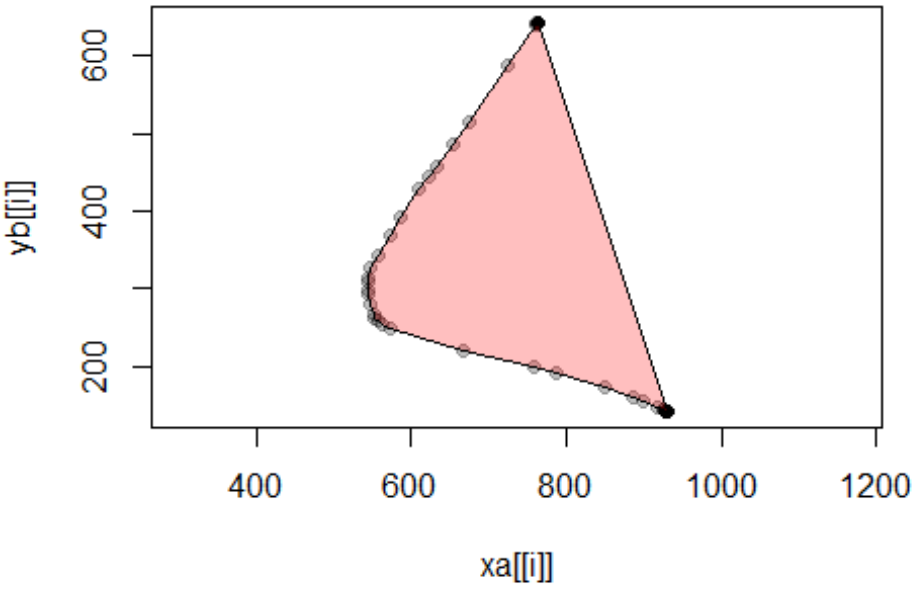

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

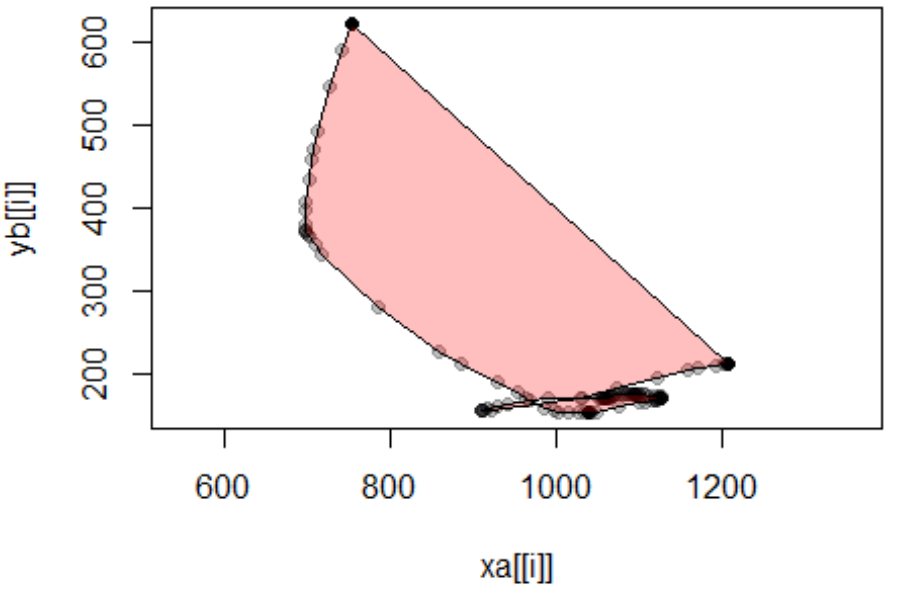

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

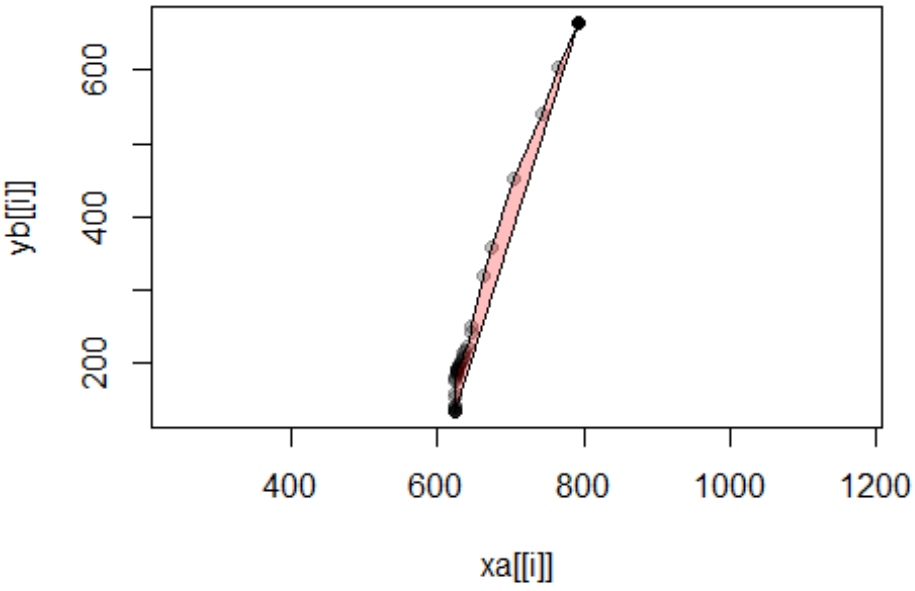

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

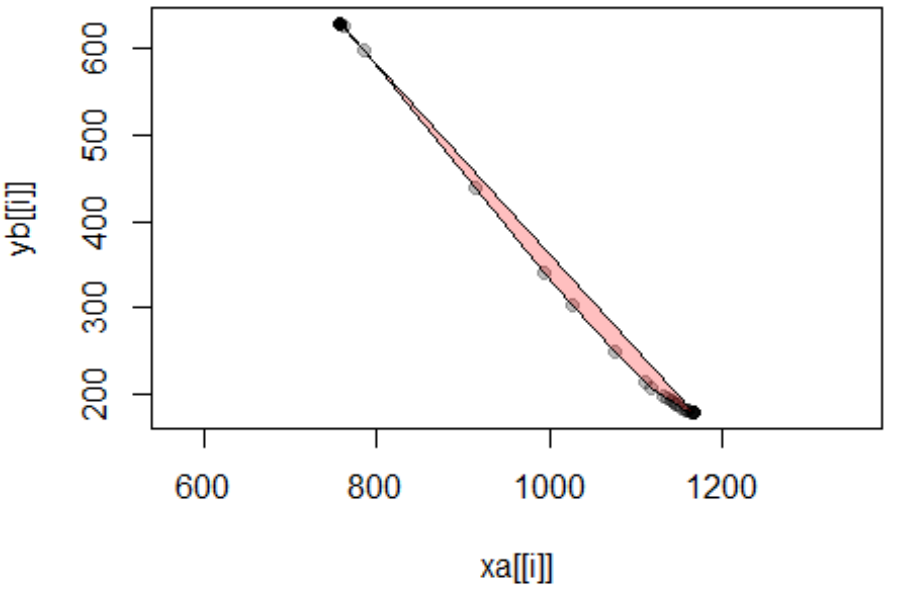

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

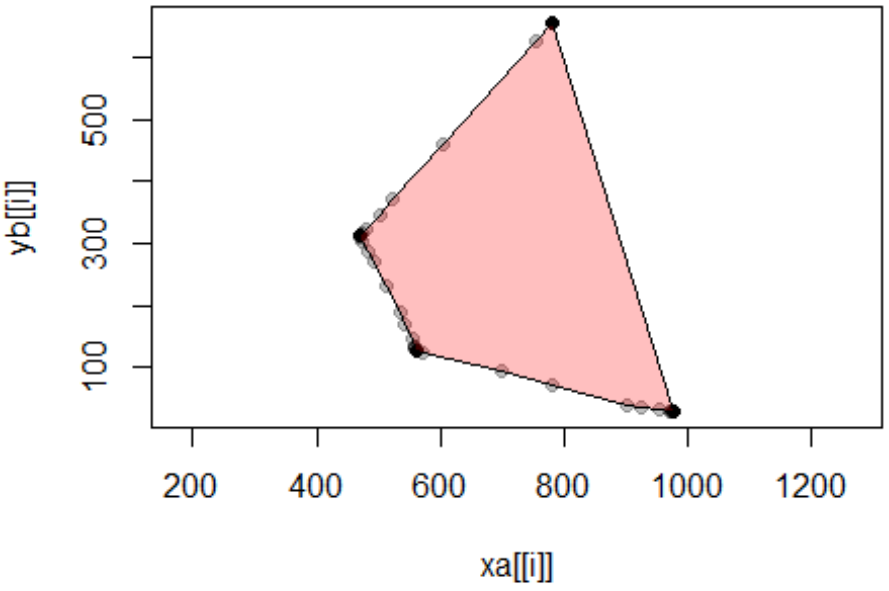

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

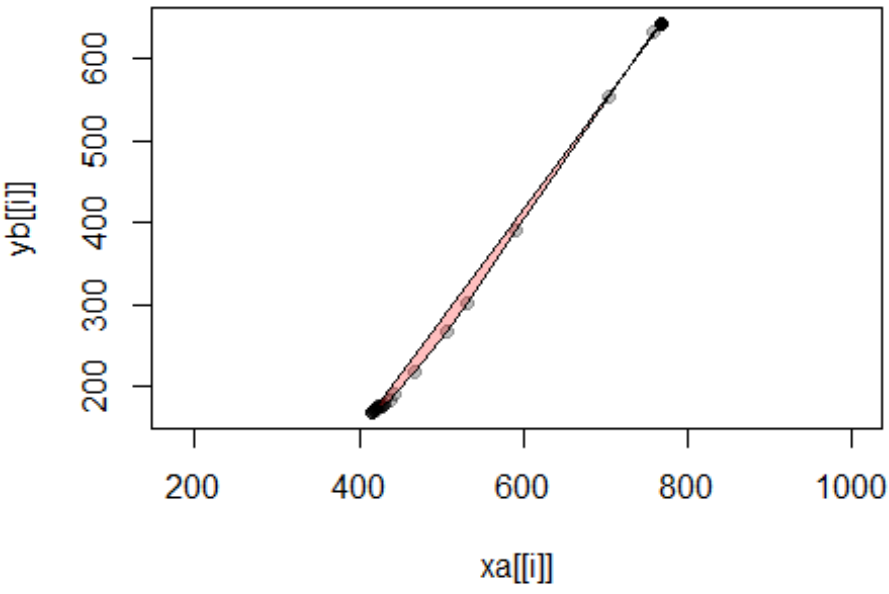

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

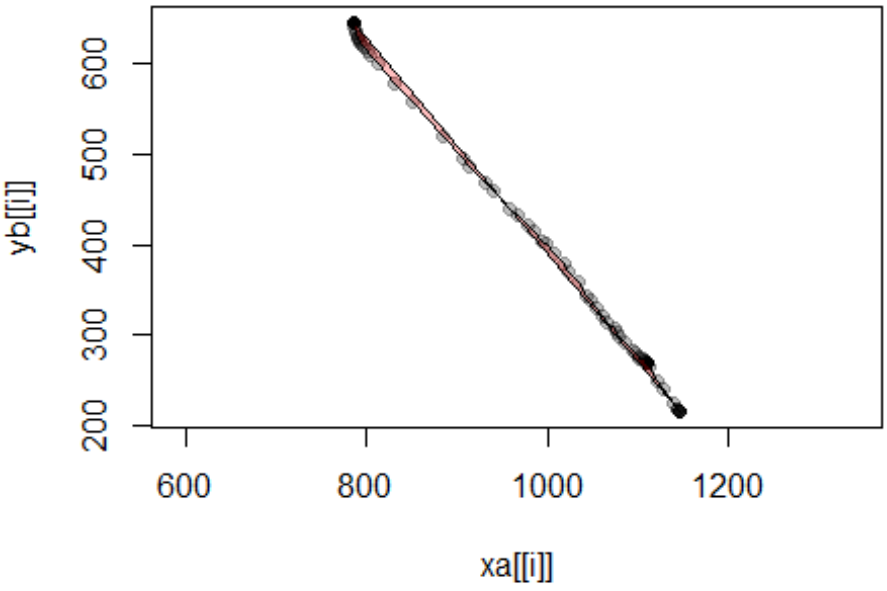

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

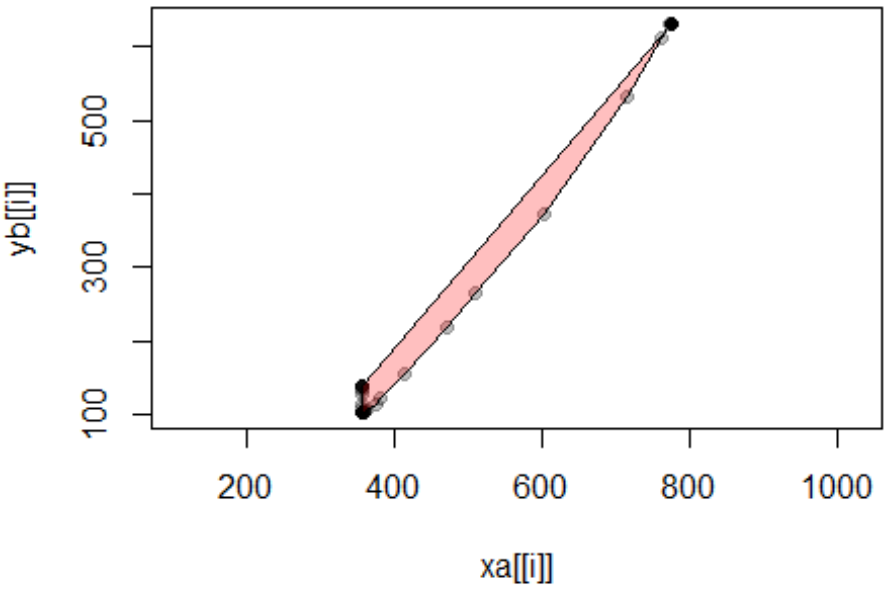

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

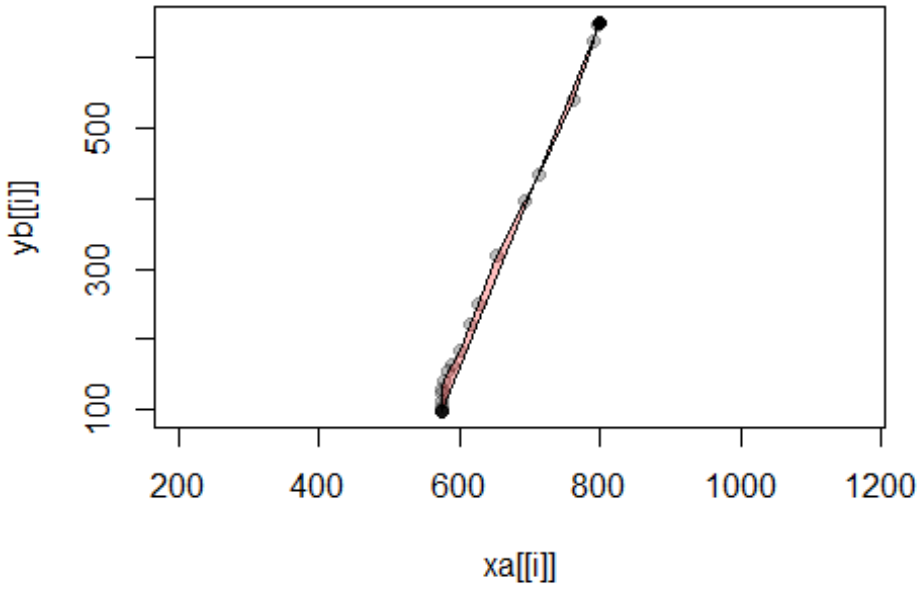

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

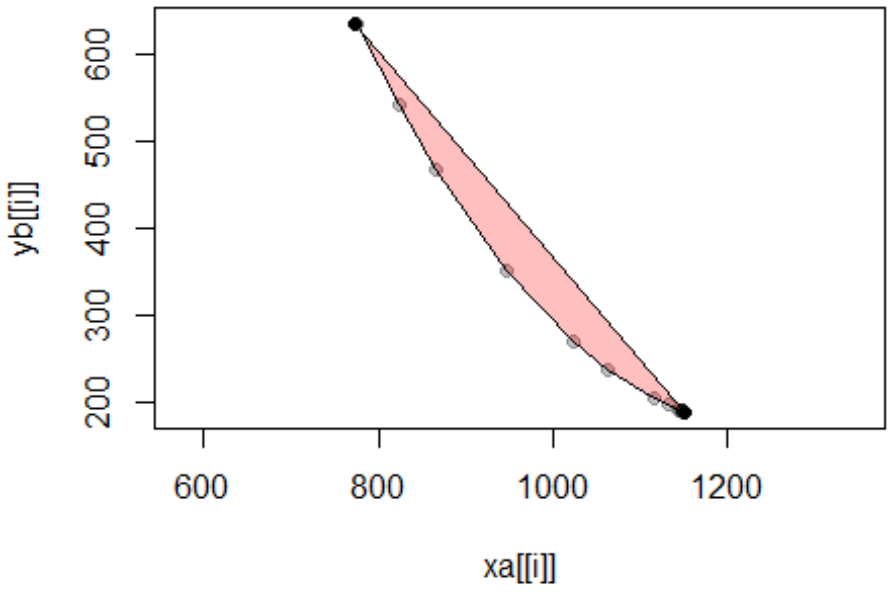

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

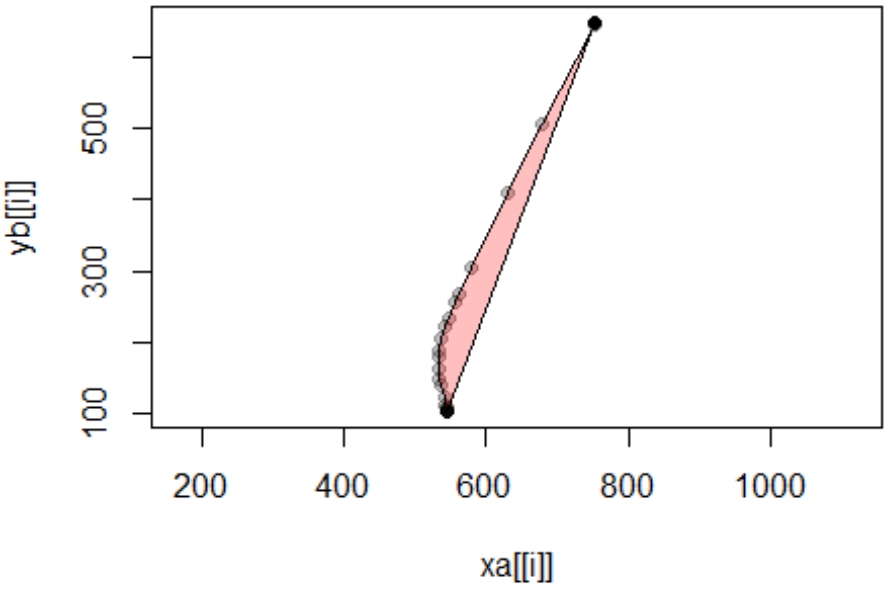

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

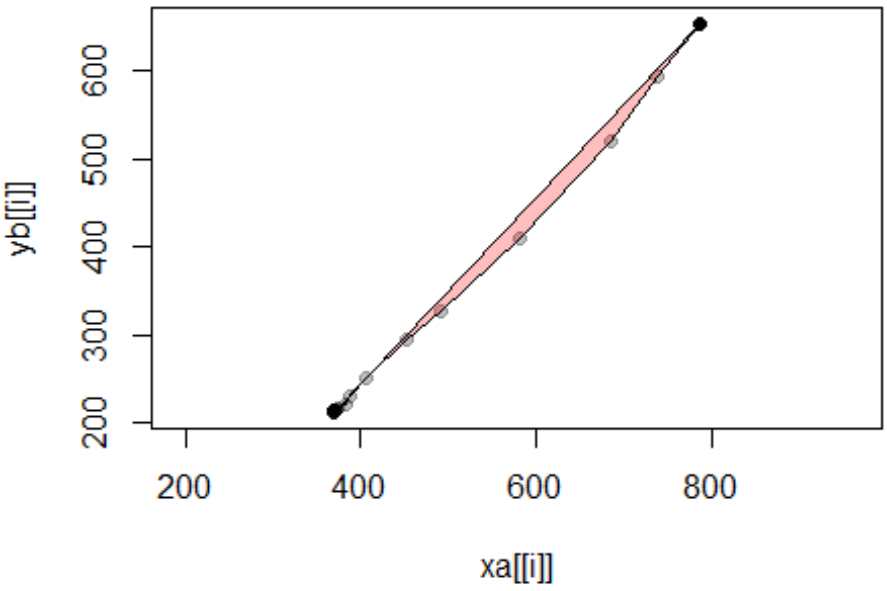

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

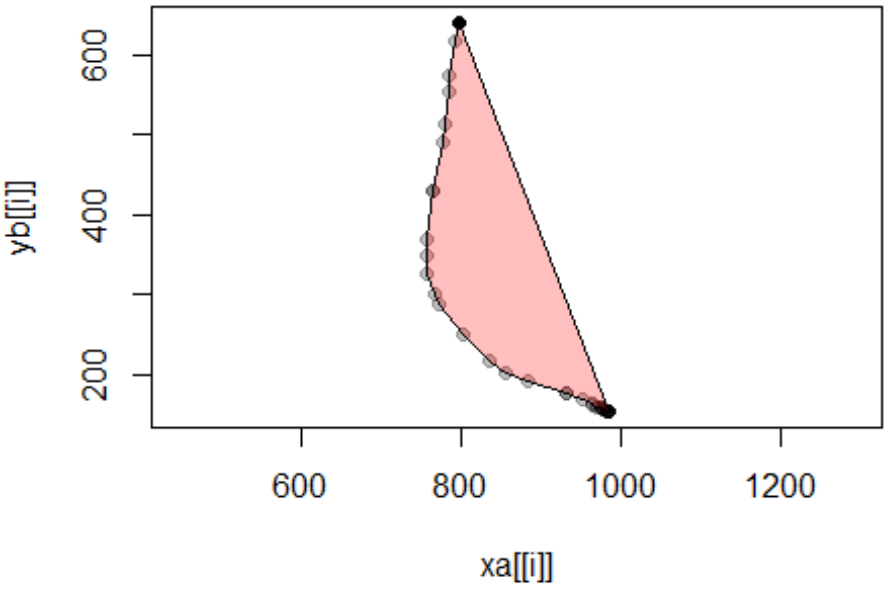

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

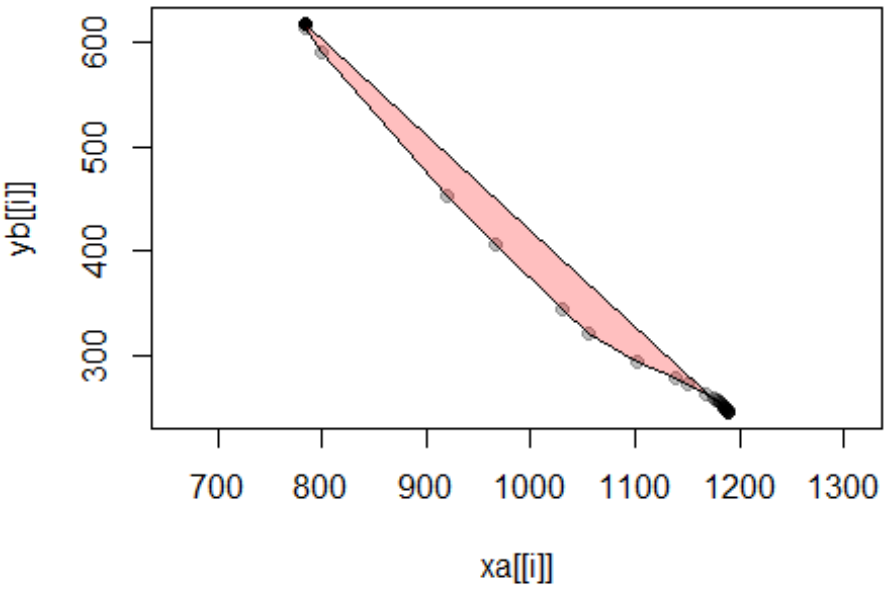

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

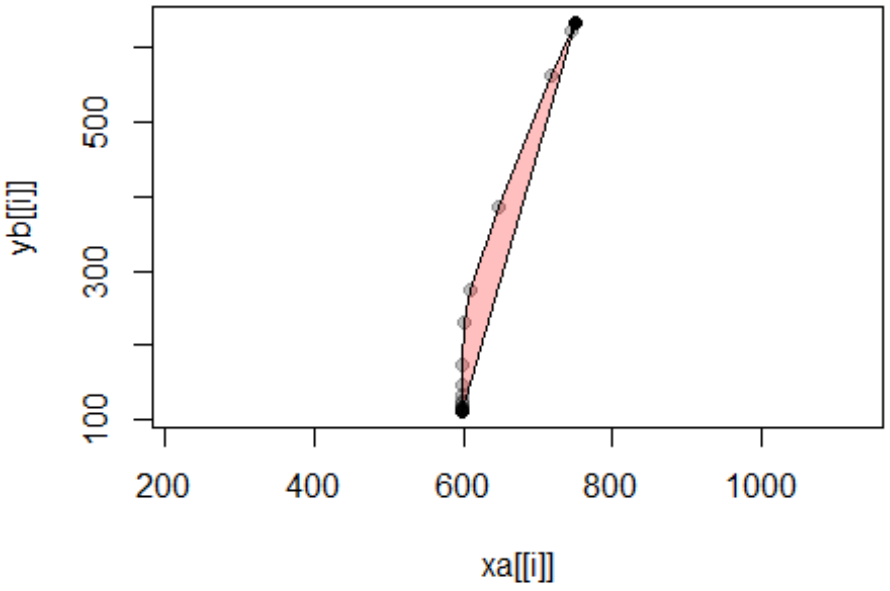

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

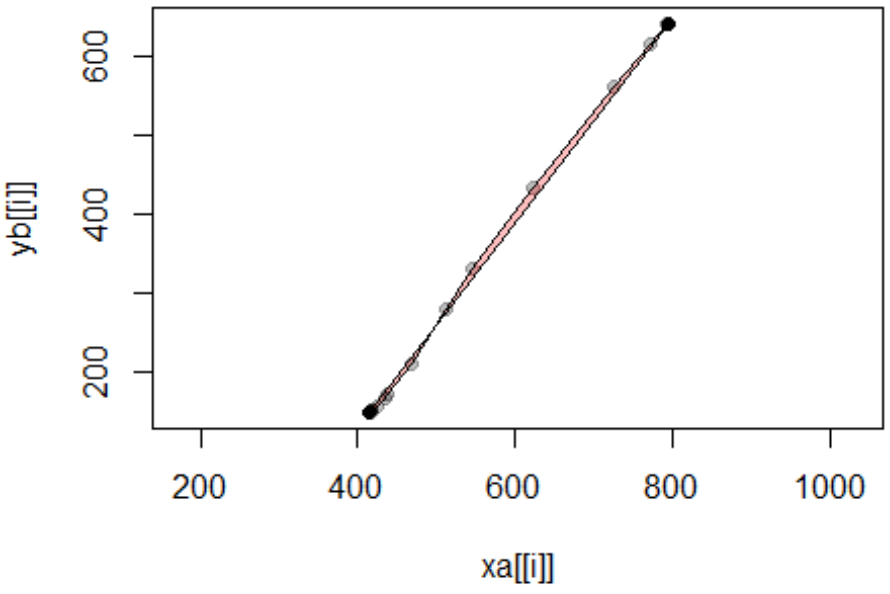

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

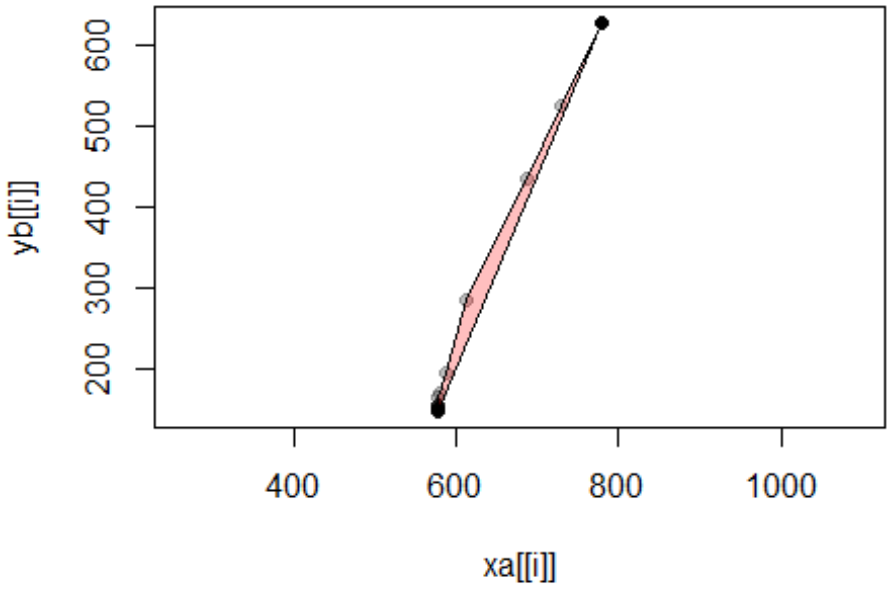

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

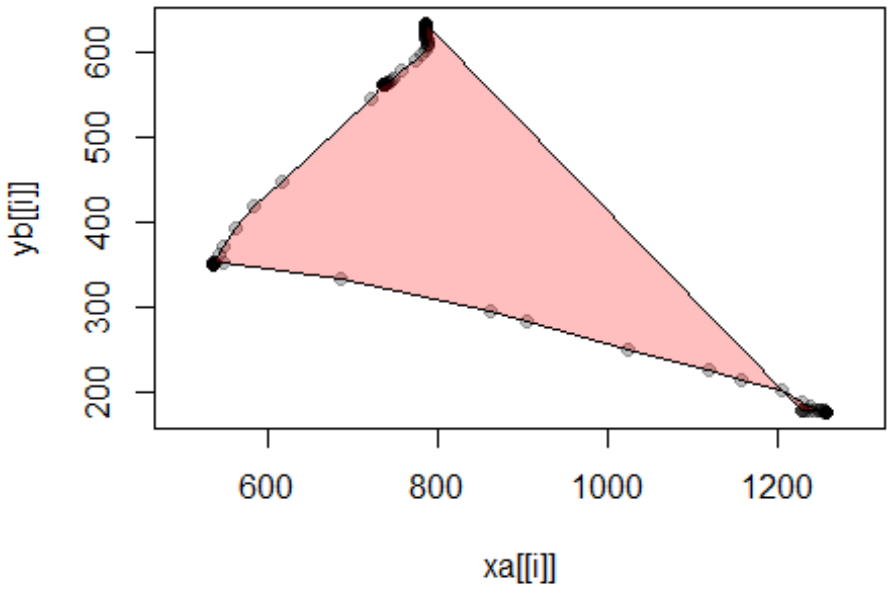

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

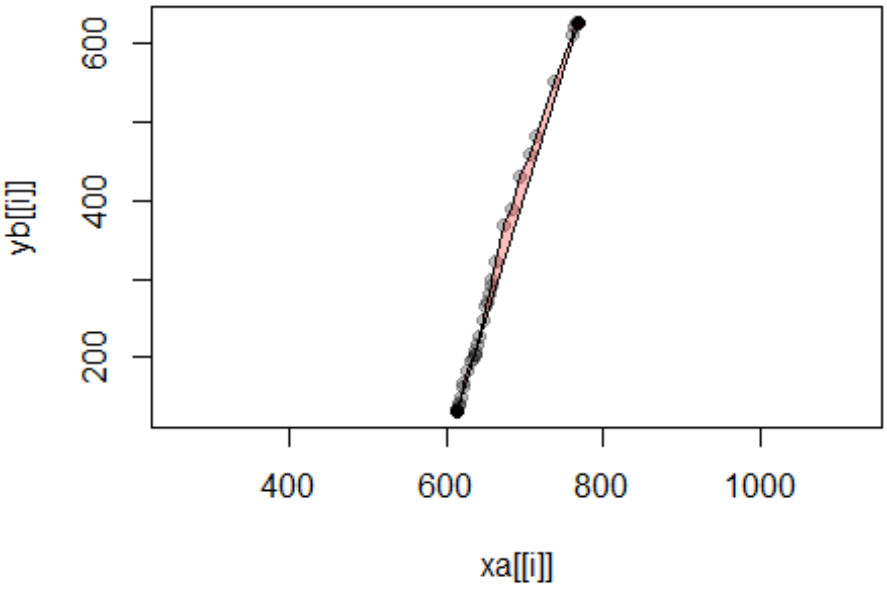

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

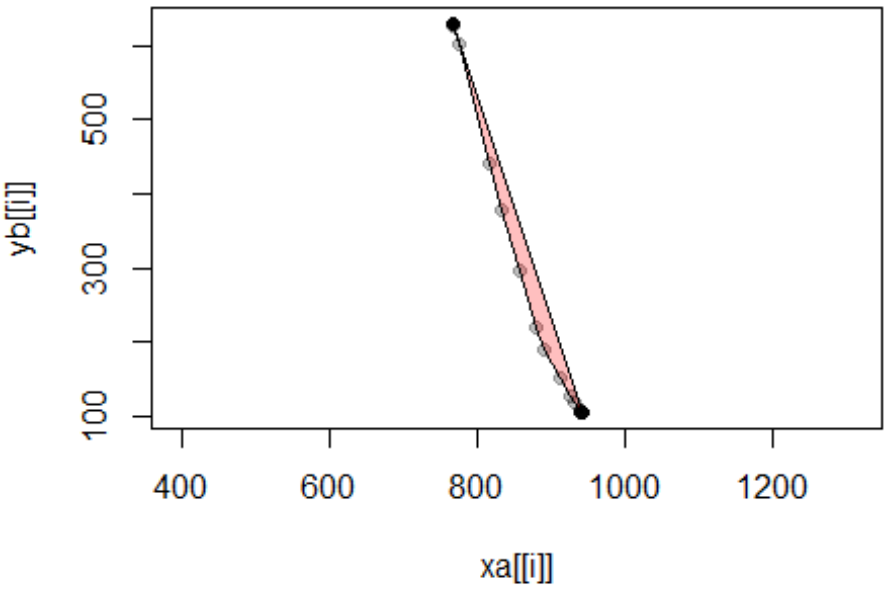

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

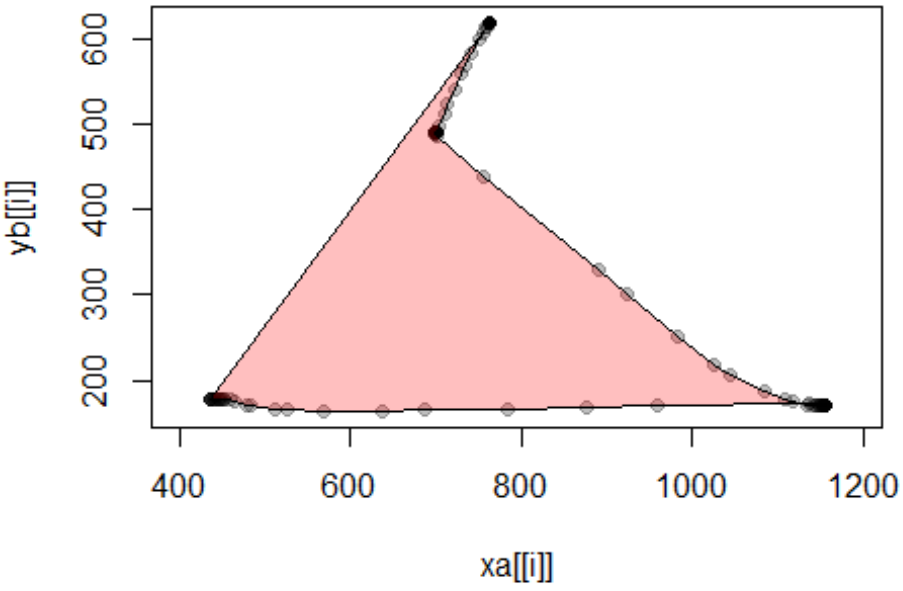

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

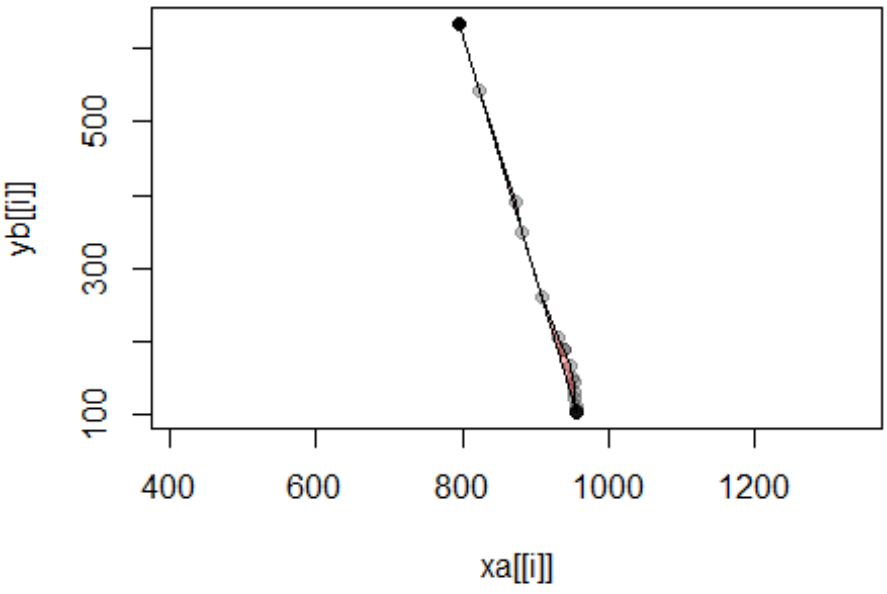

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

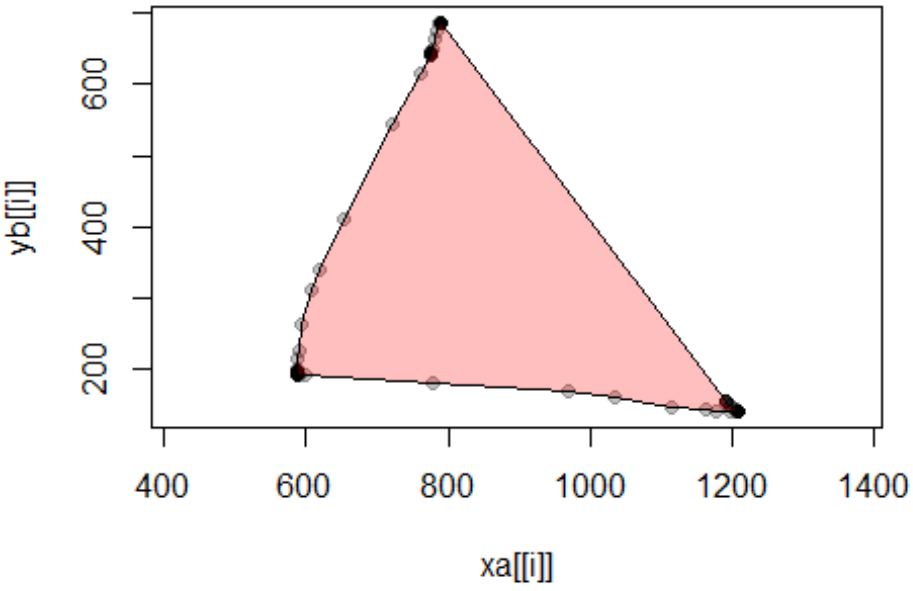

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

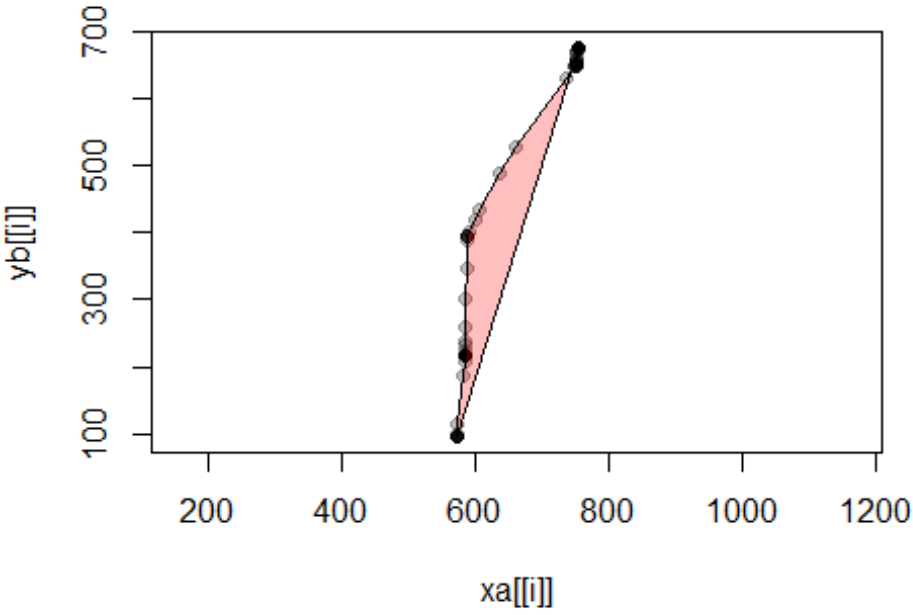

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

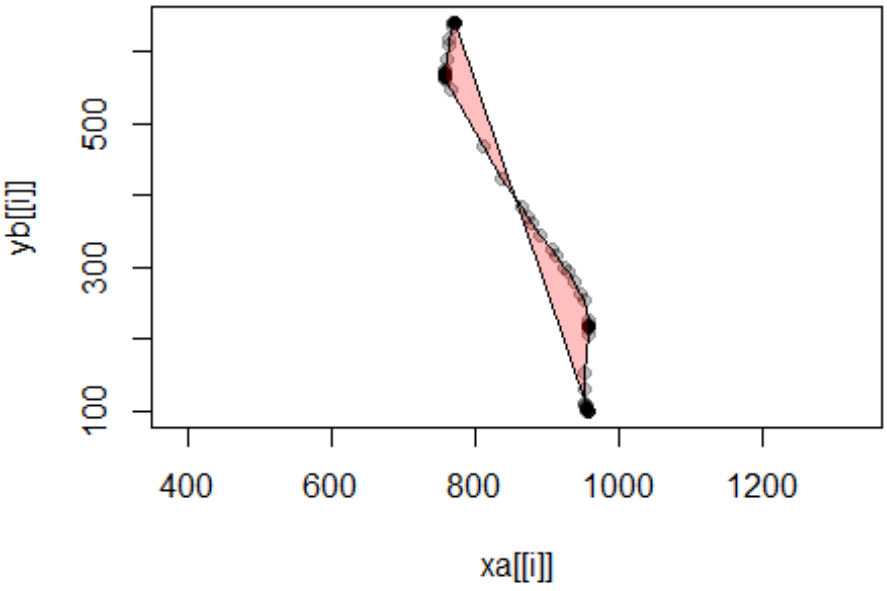

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

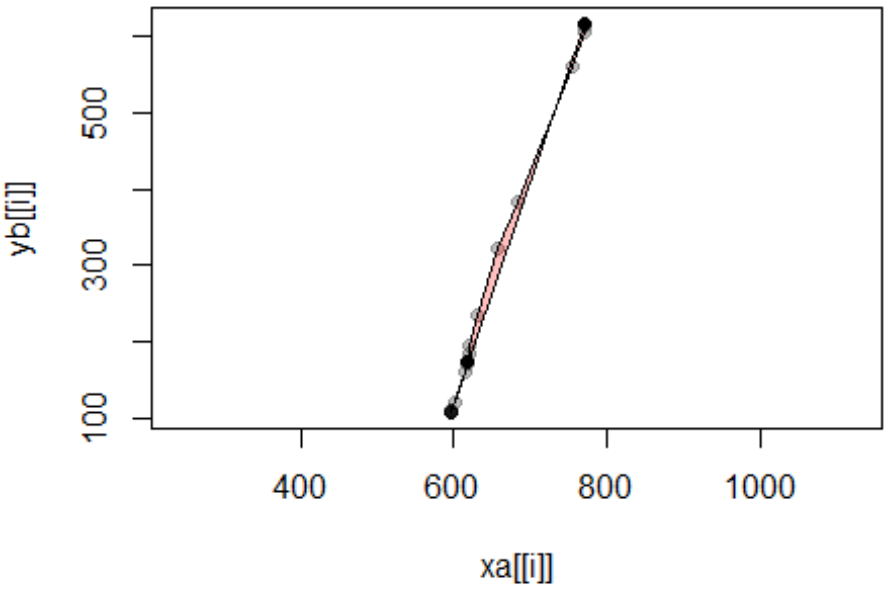

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

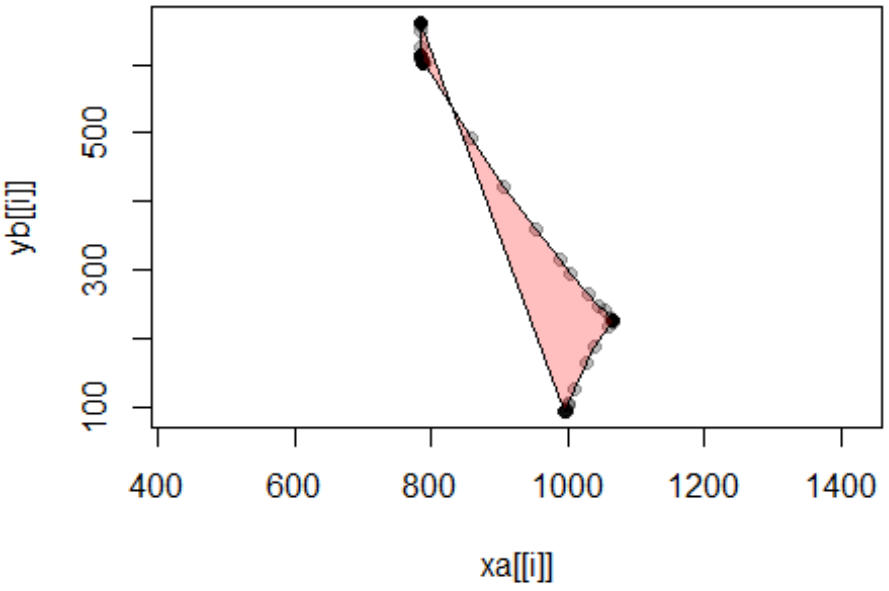

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

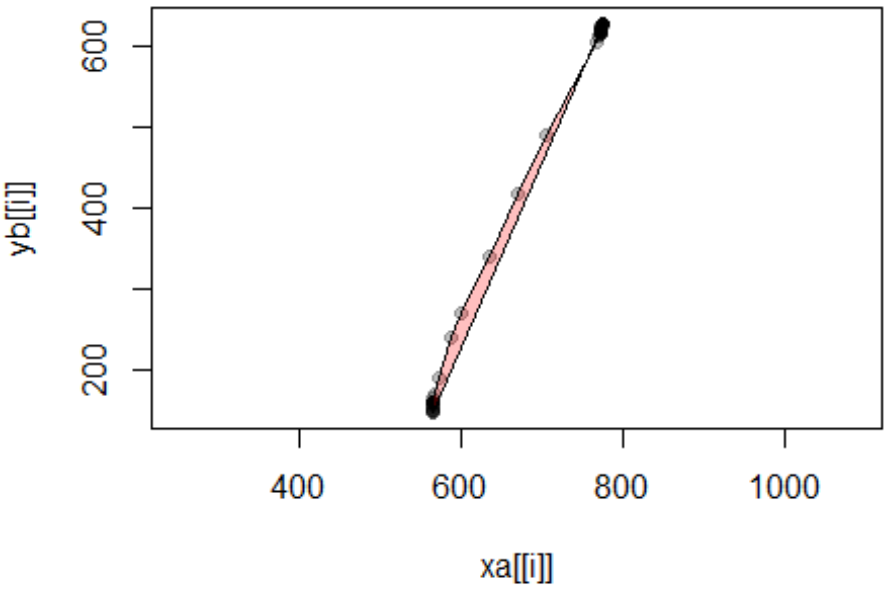

yb[[i]]
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

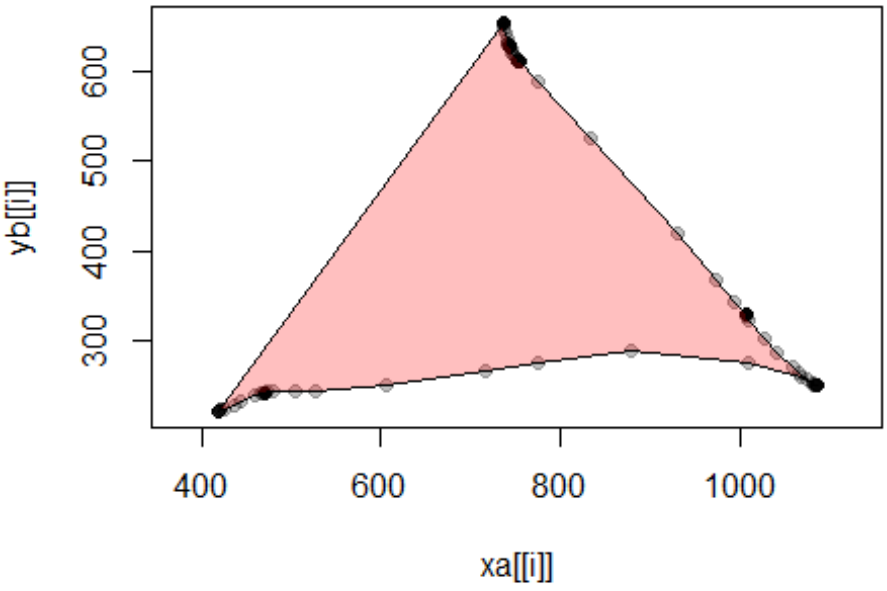

yb[[i]]
100
300
500
700
200
400
600
800
1000
1200
xa[[i]]

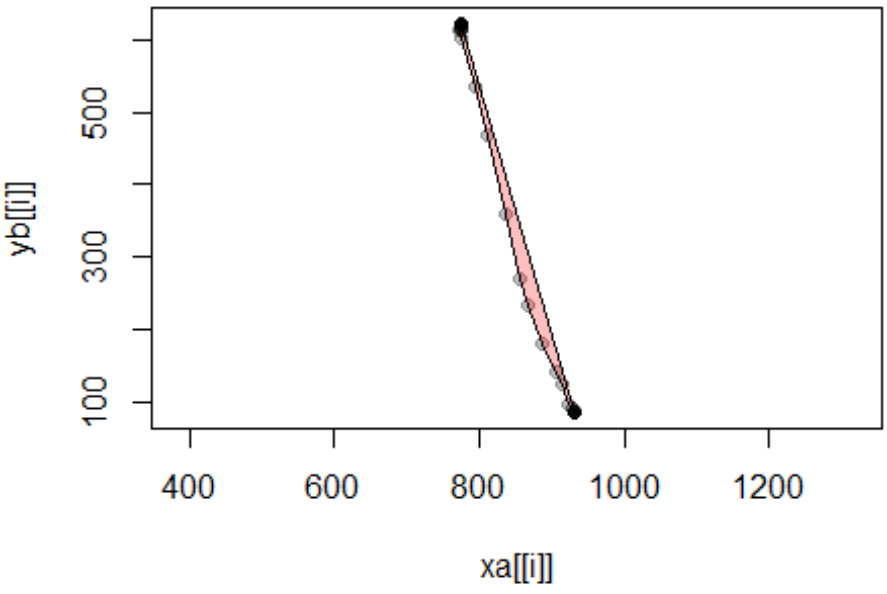

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

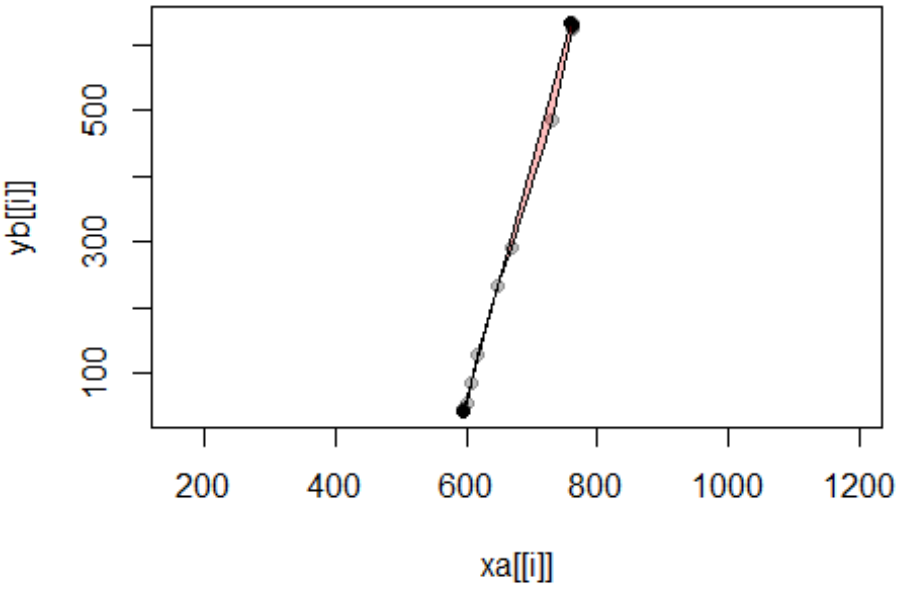

yb[[i]]
100
300
500
400
600
800
1000
xa[[i]]

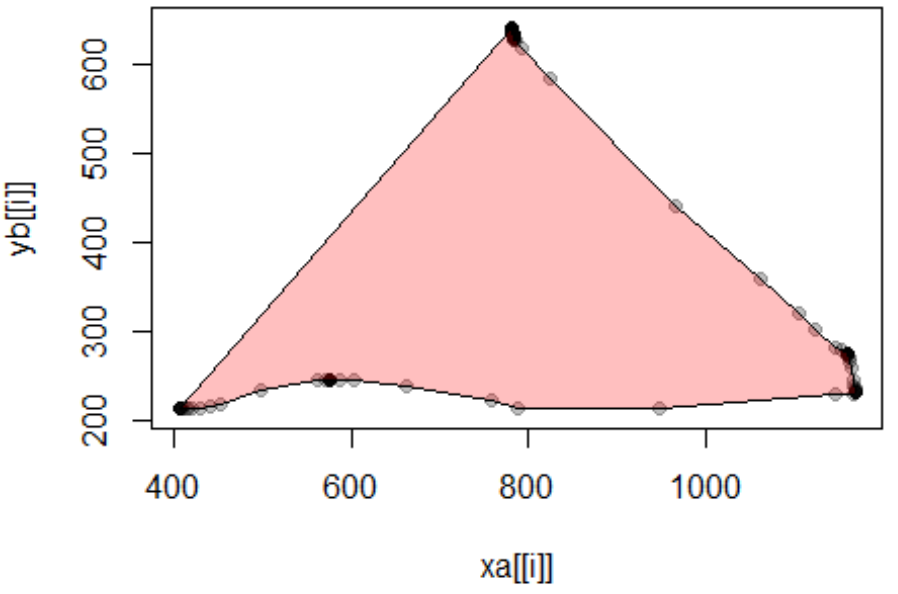

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

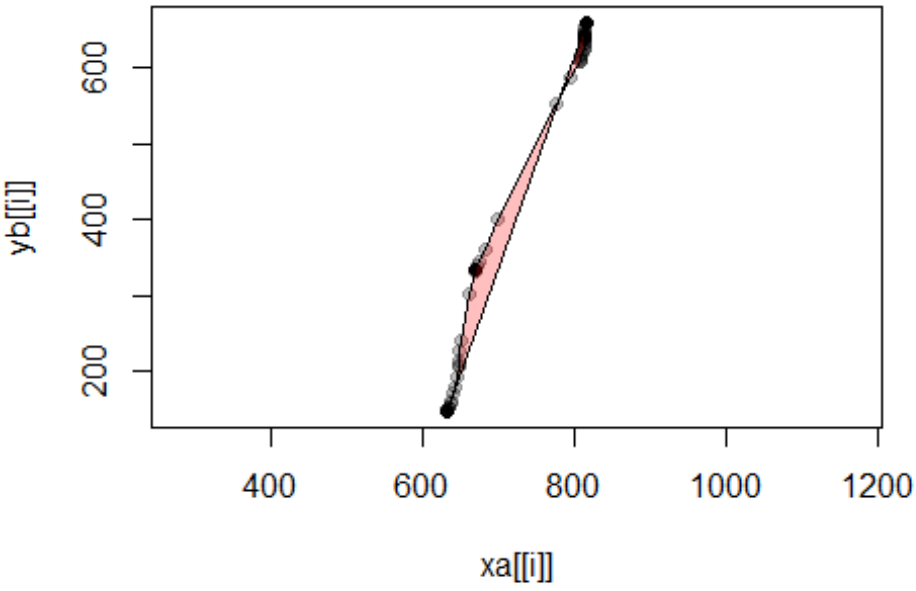

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

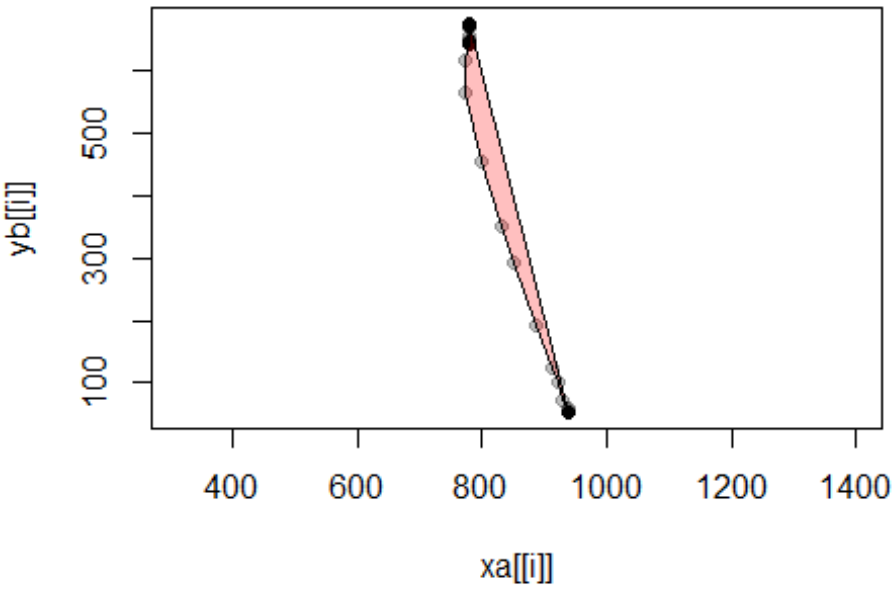

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

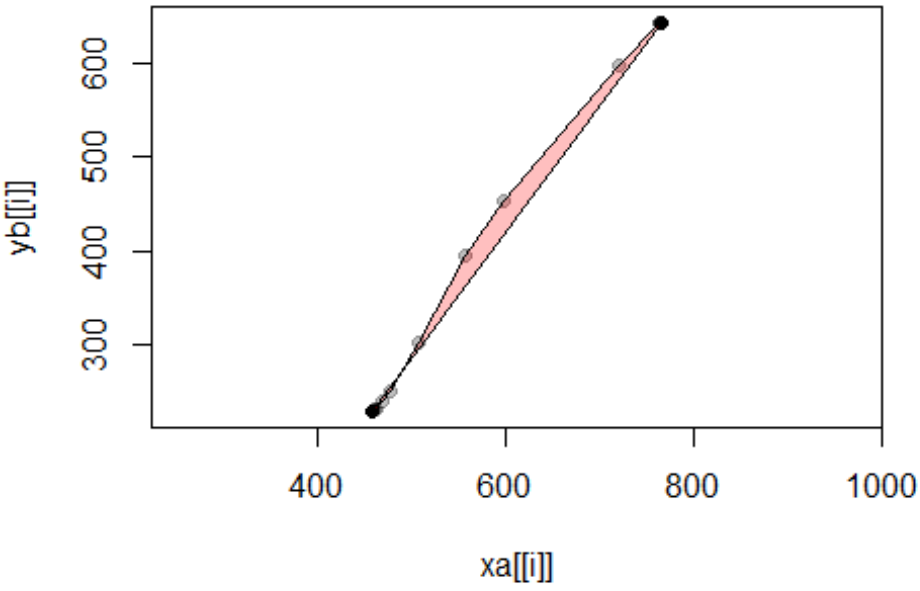

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

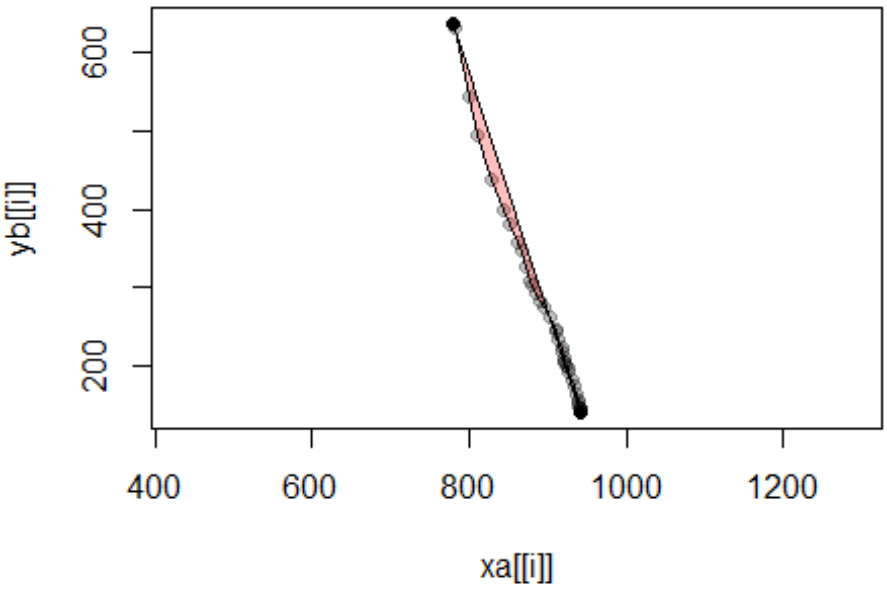

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

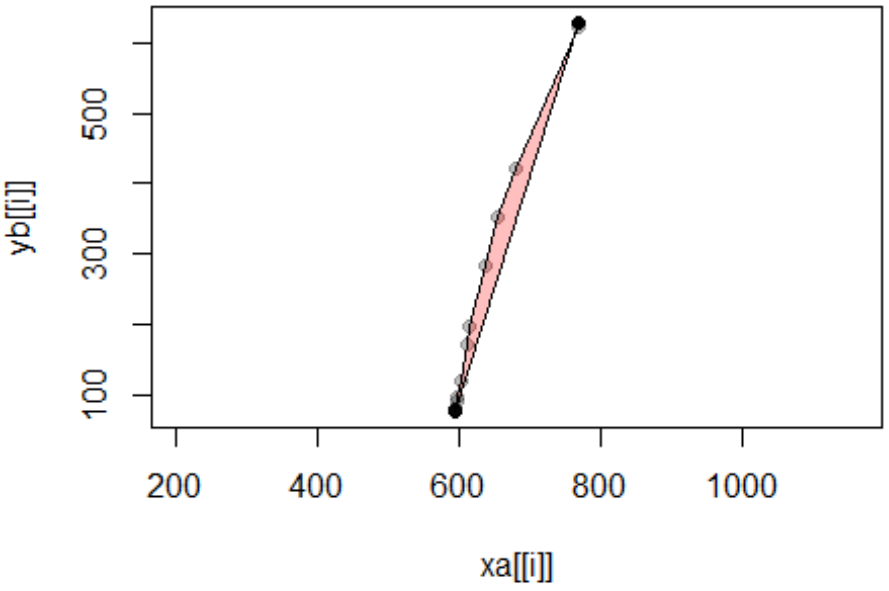

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

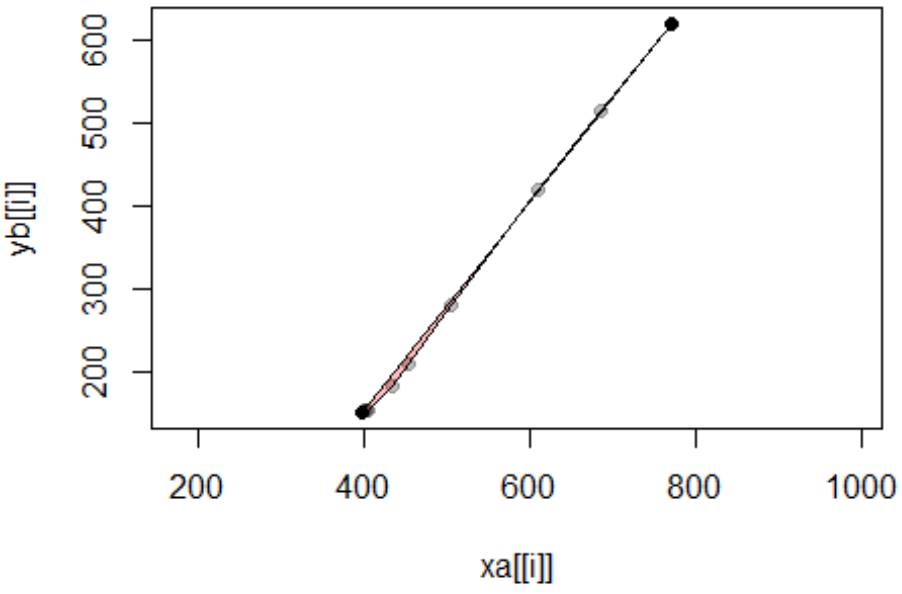

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

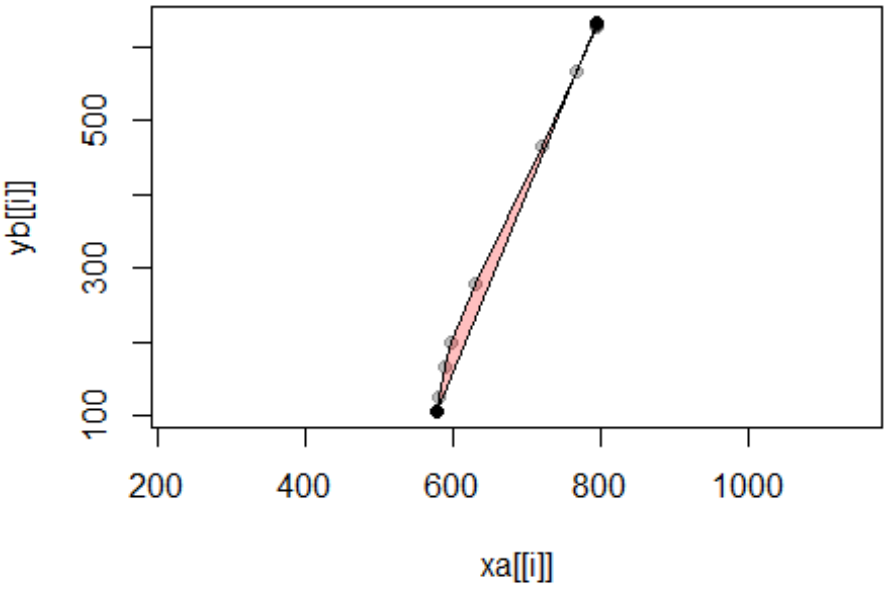

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

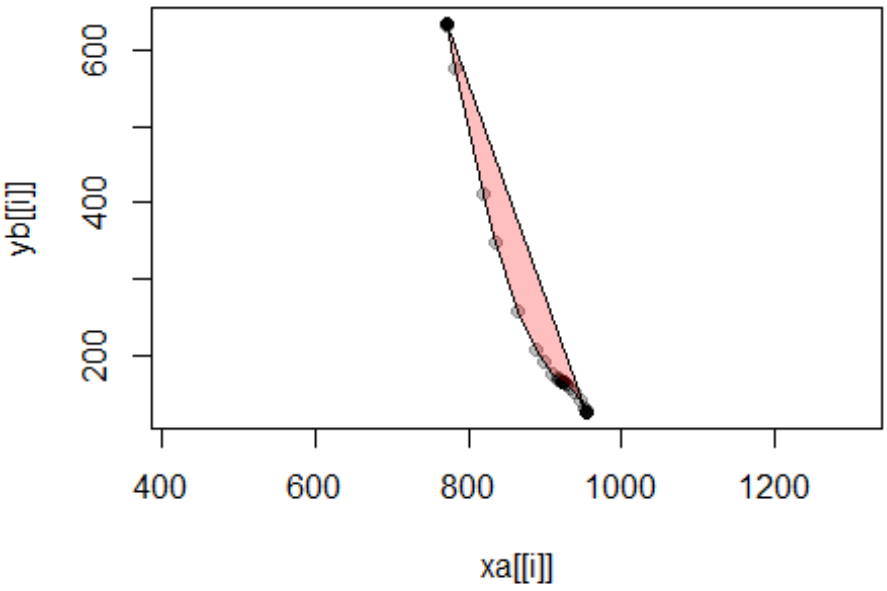

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

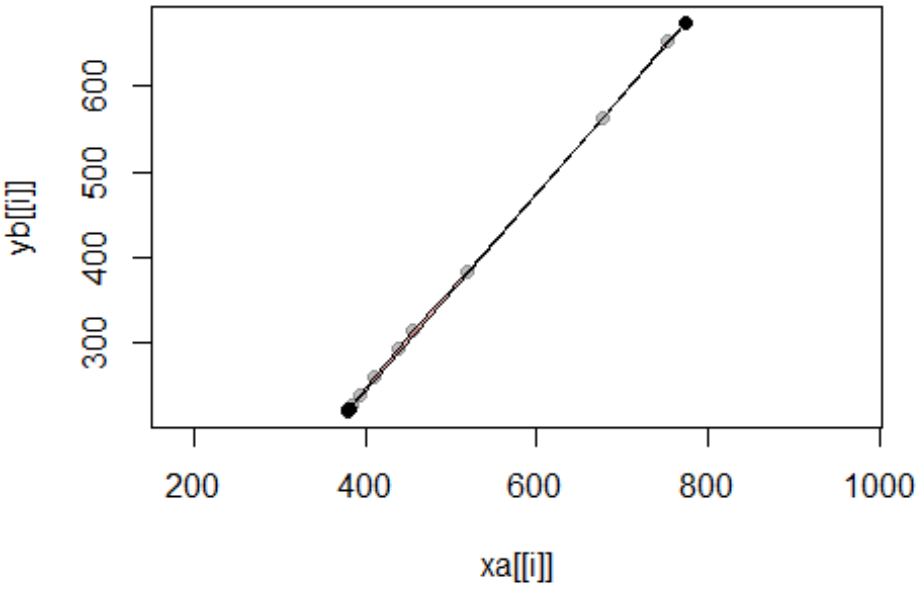

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

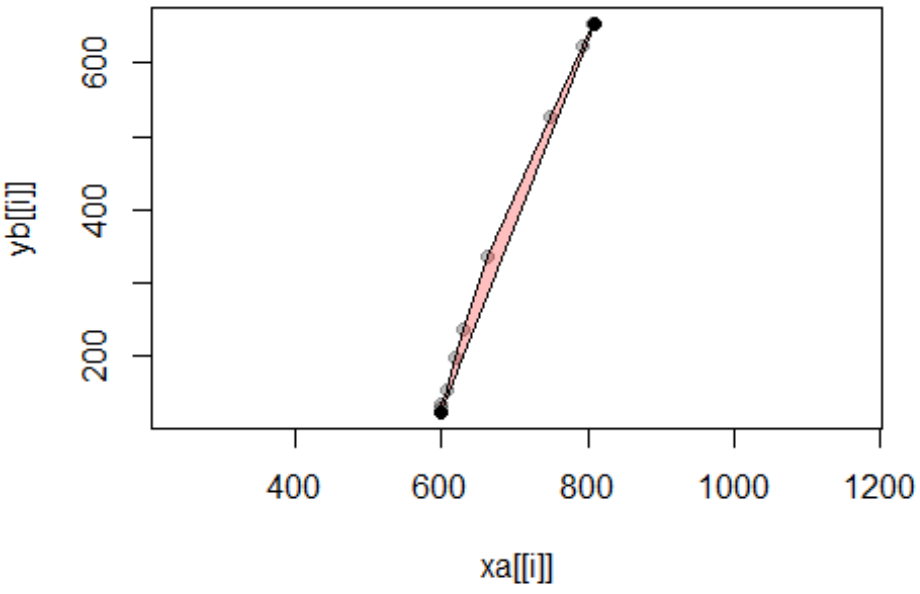

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

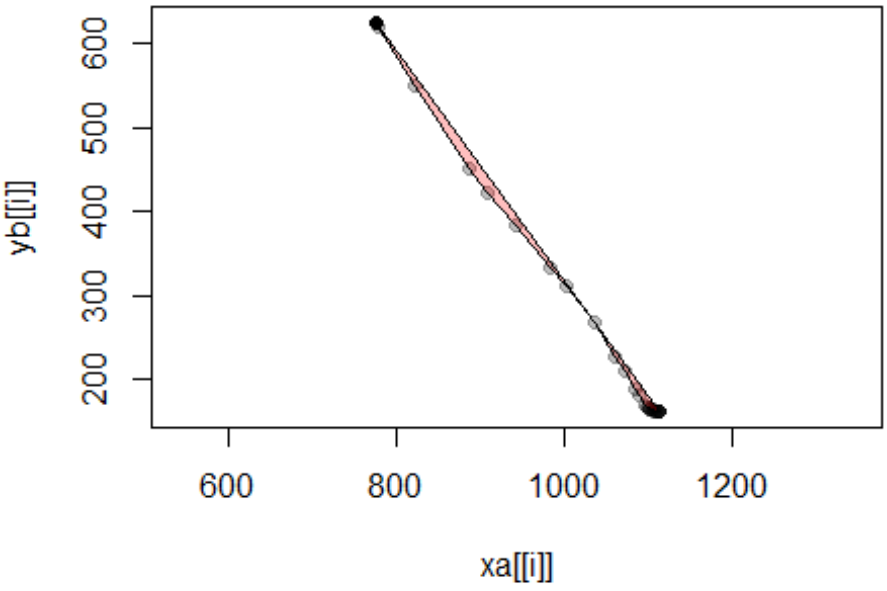

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

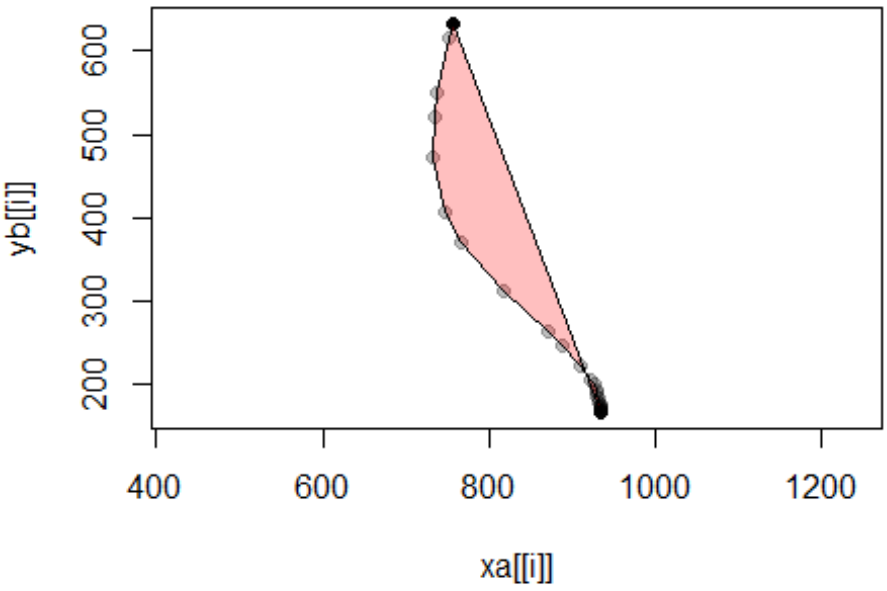

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

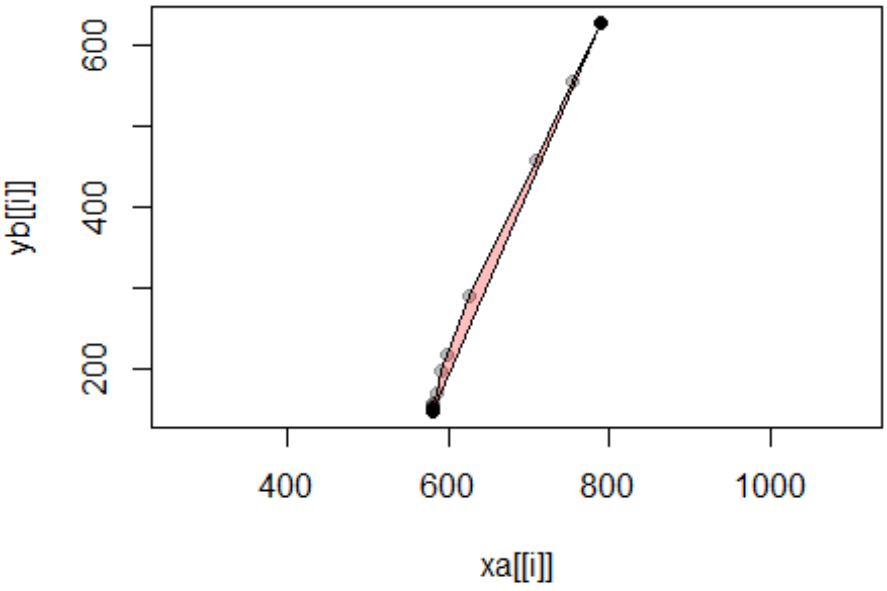

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

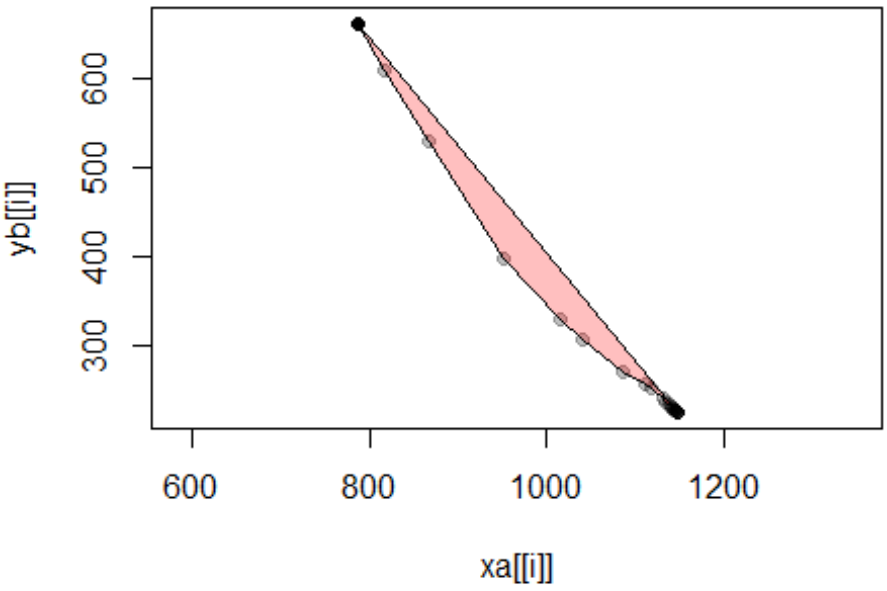

yb[[i]]
300
400
500
600
200
400
600
800
1000
xa[[i]]

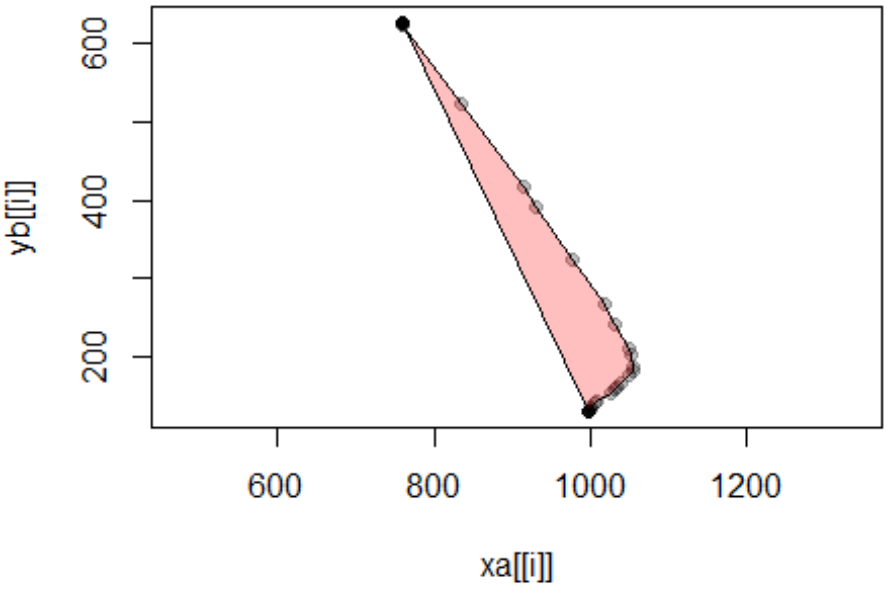

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

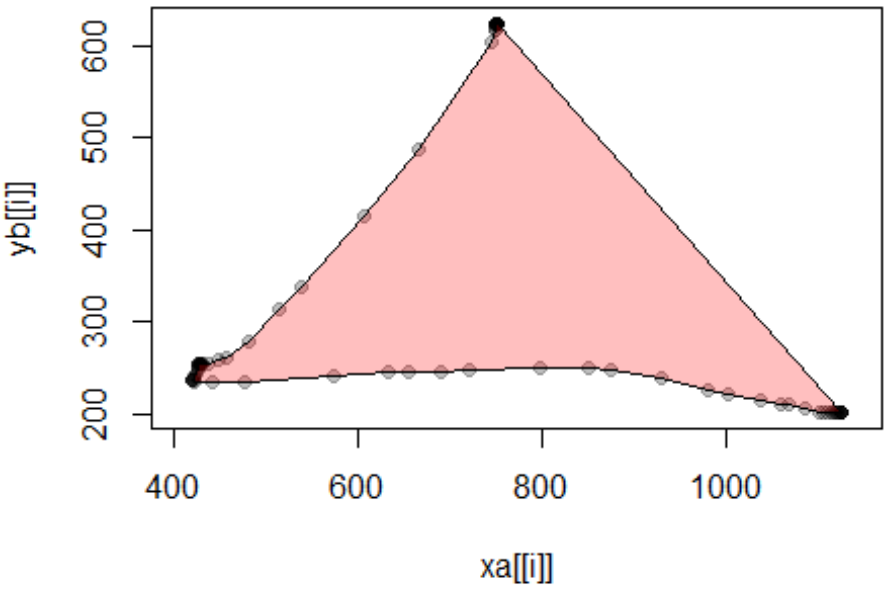

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

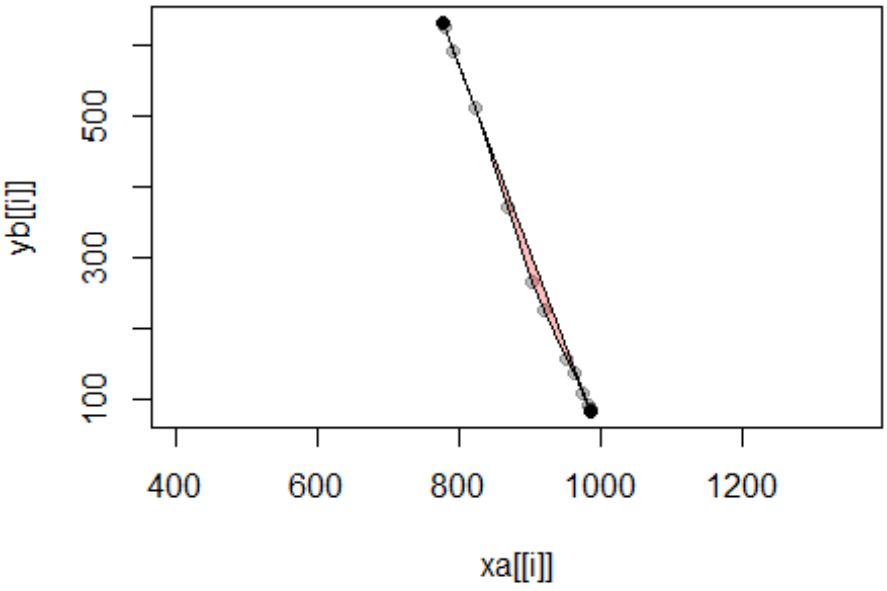

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

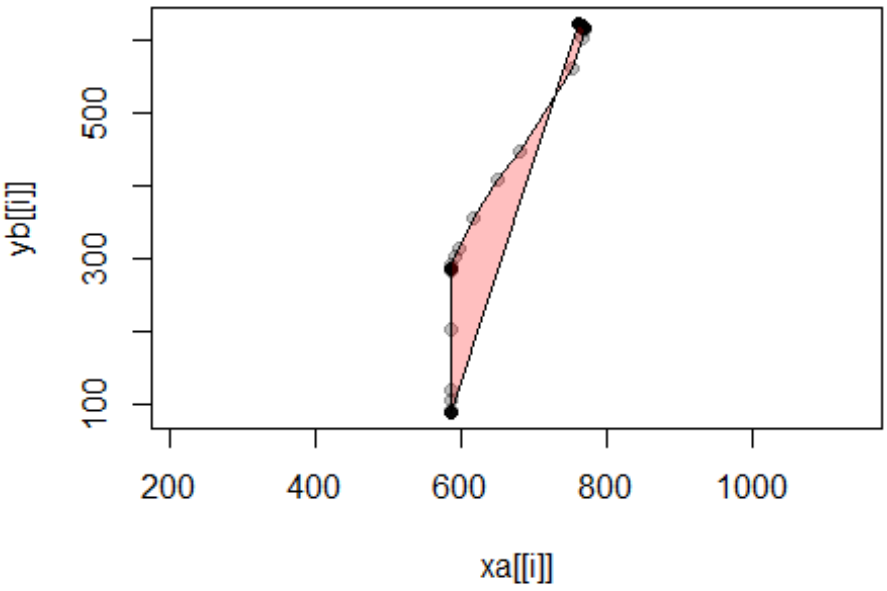

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

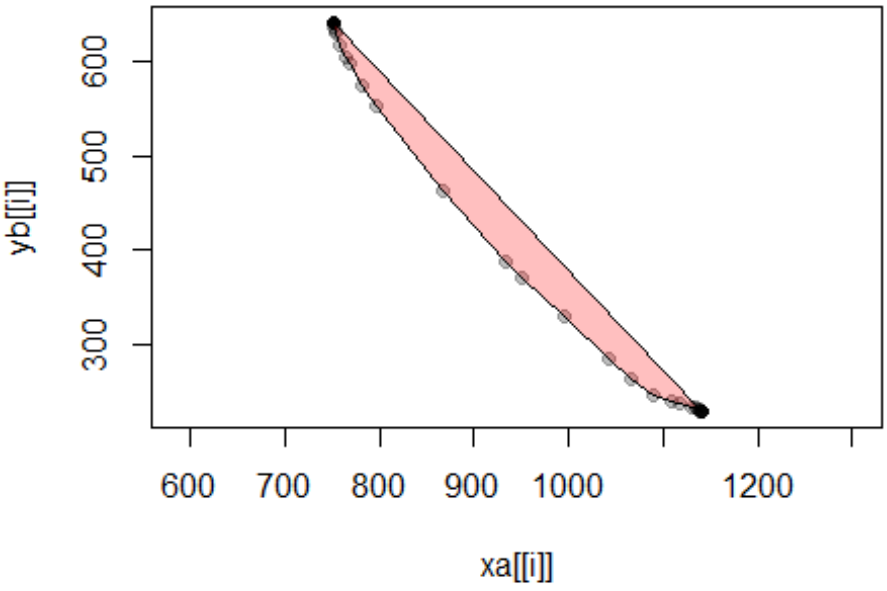

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

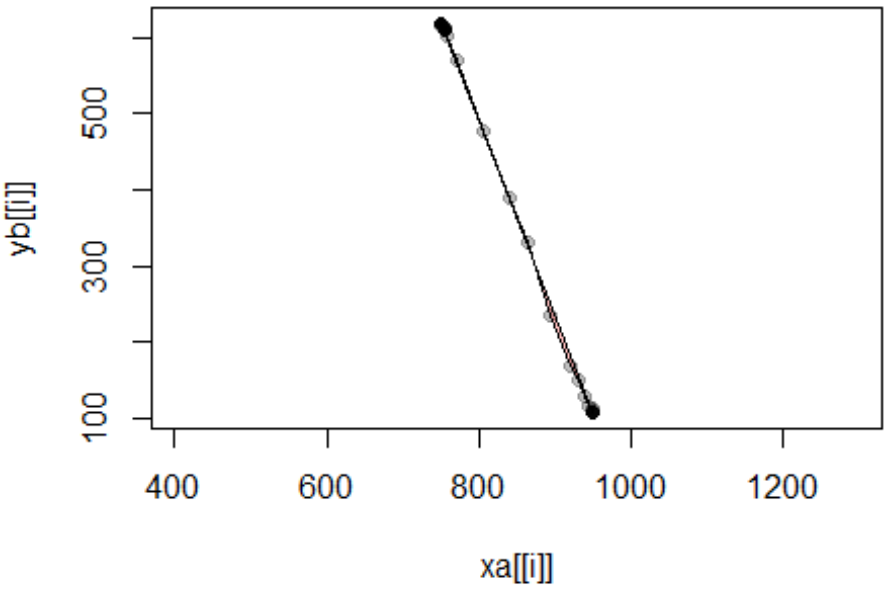

yb[[i]]
600
400
200
600
800
1000
1200
xa[[i]]

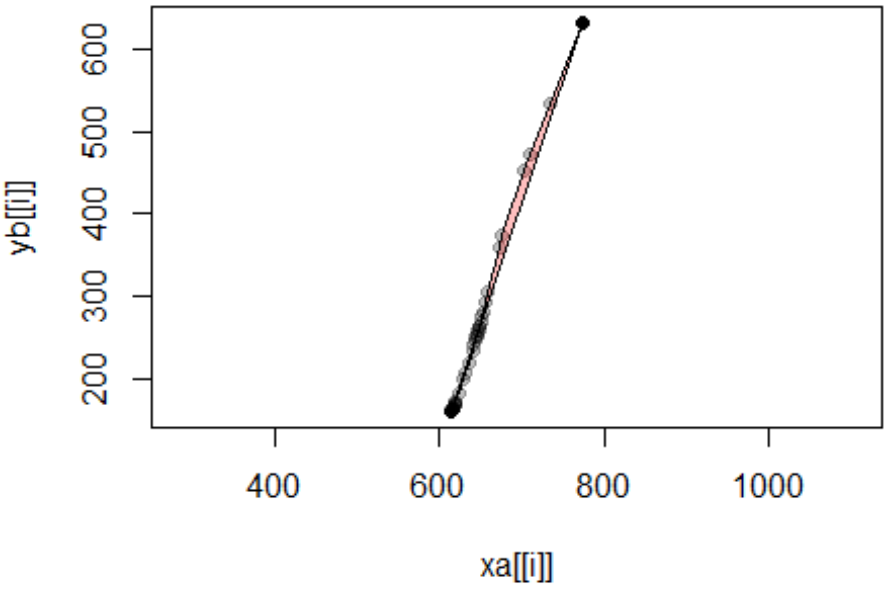

yb[[i]]
600
500
400
300
200
400
600
800
1000
xa[[i]]

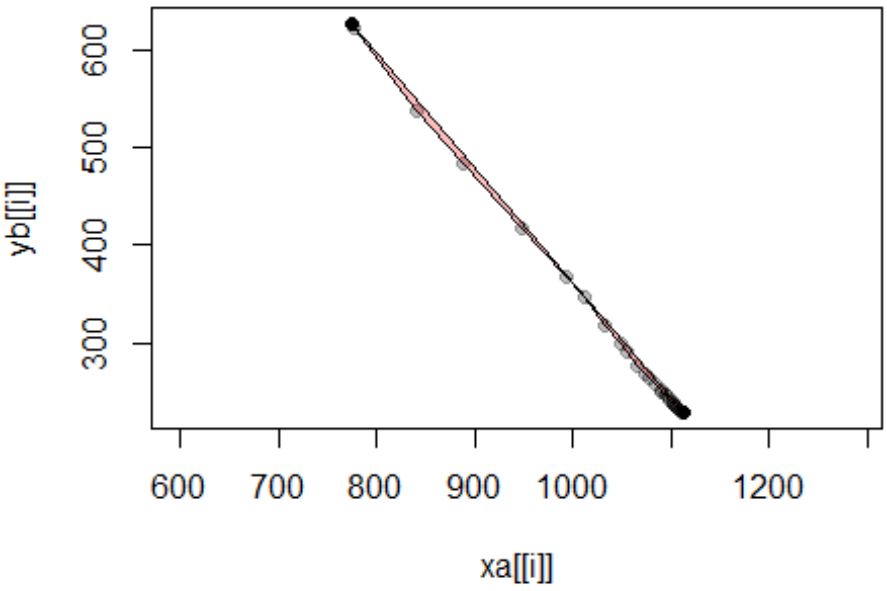

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

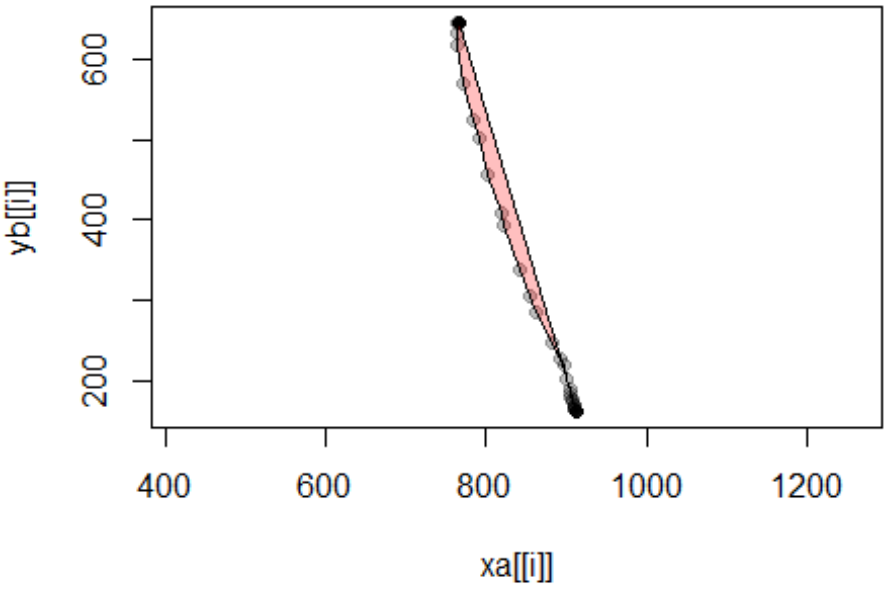

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

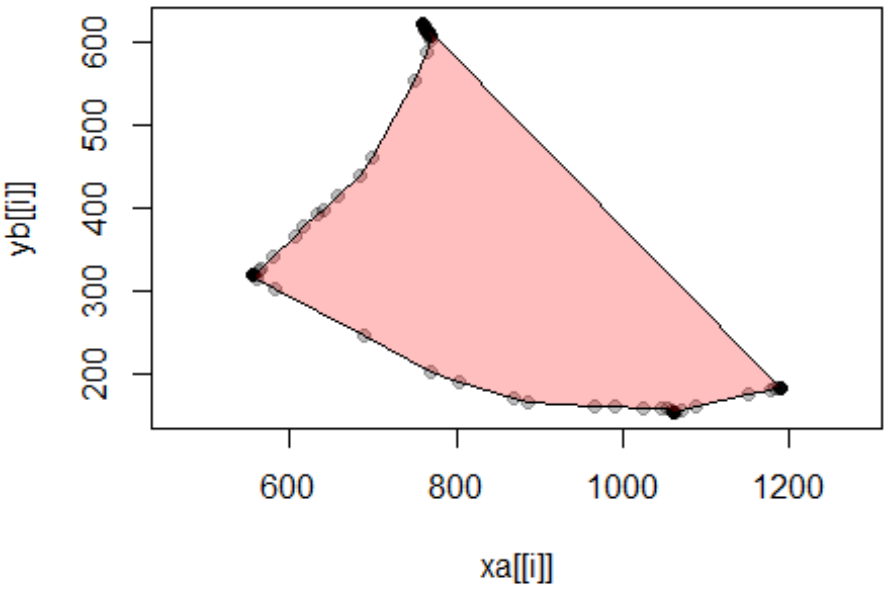

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

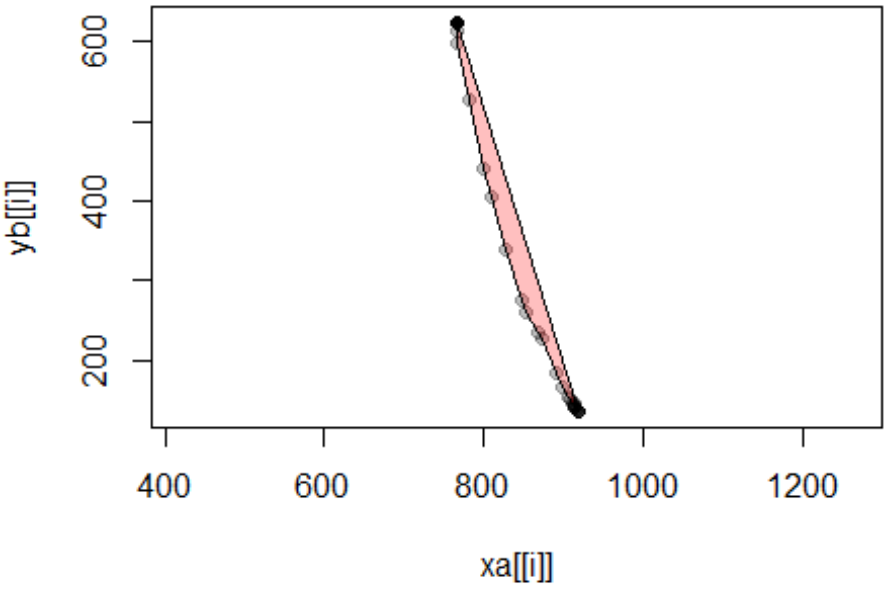

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

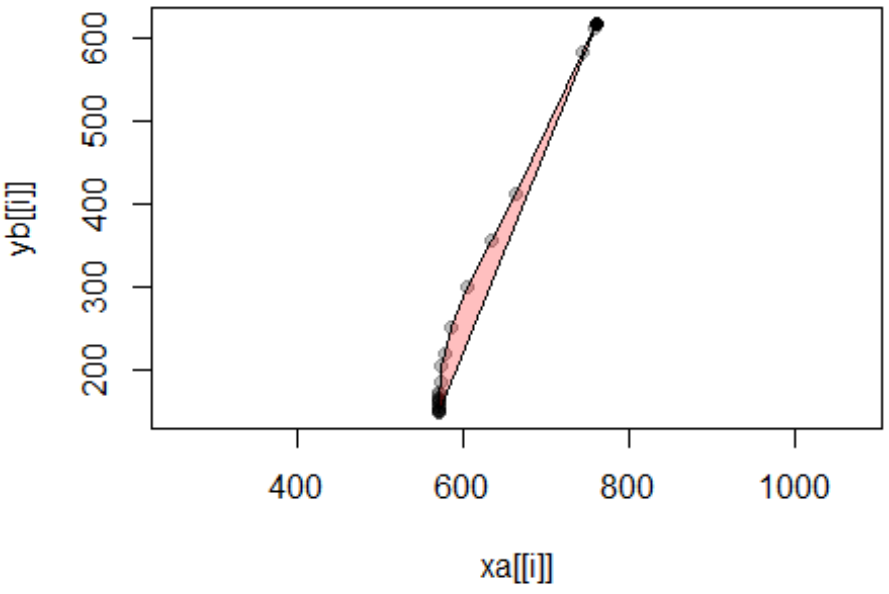

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

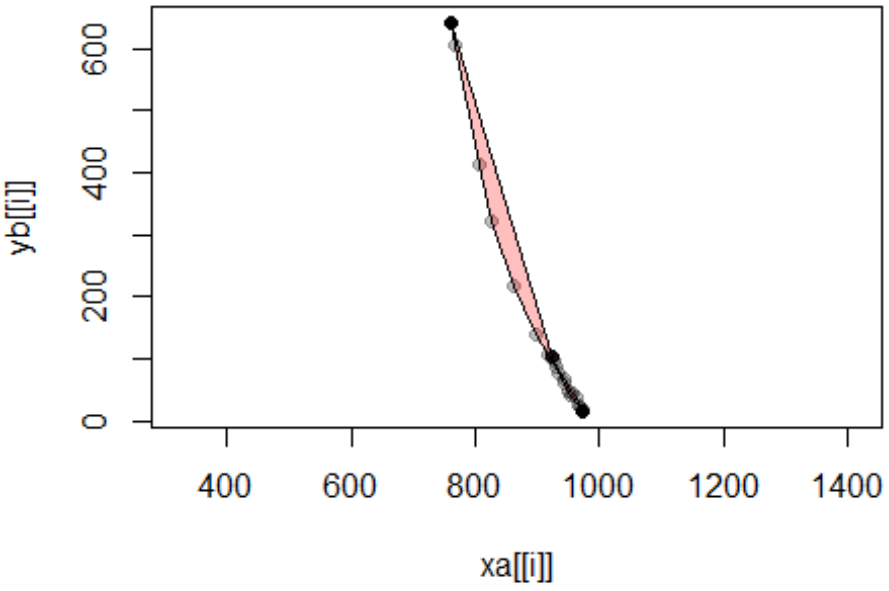

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

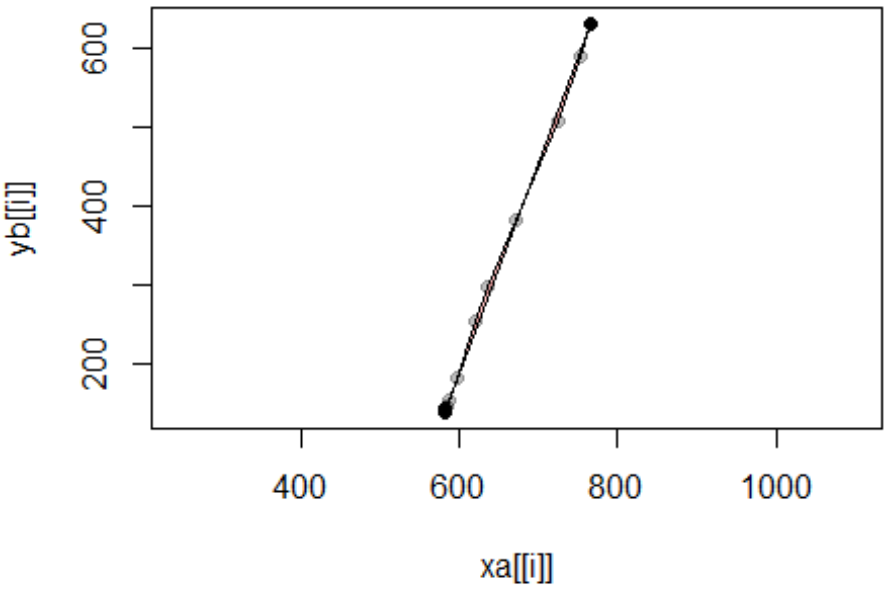

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

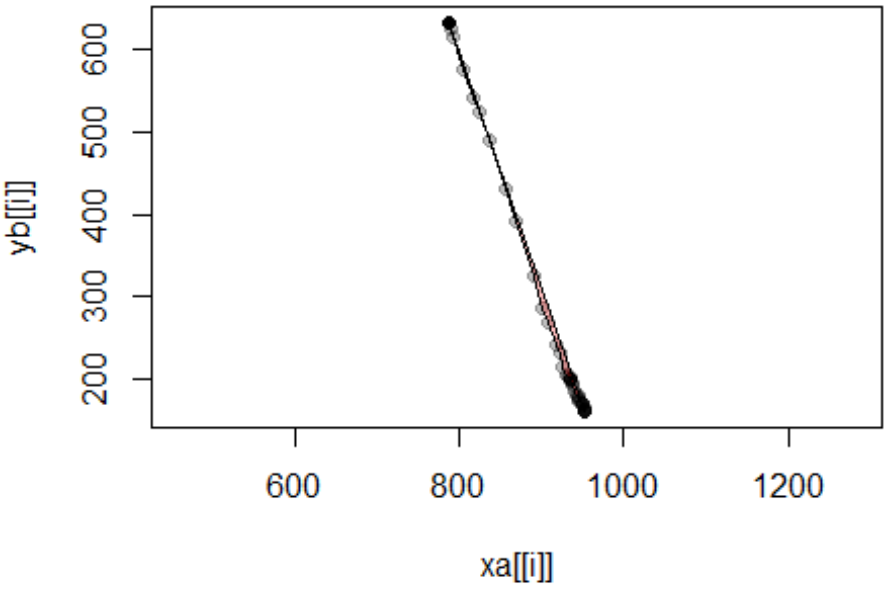

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

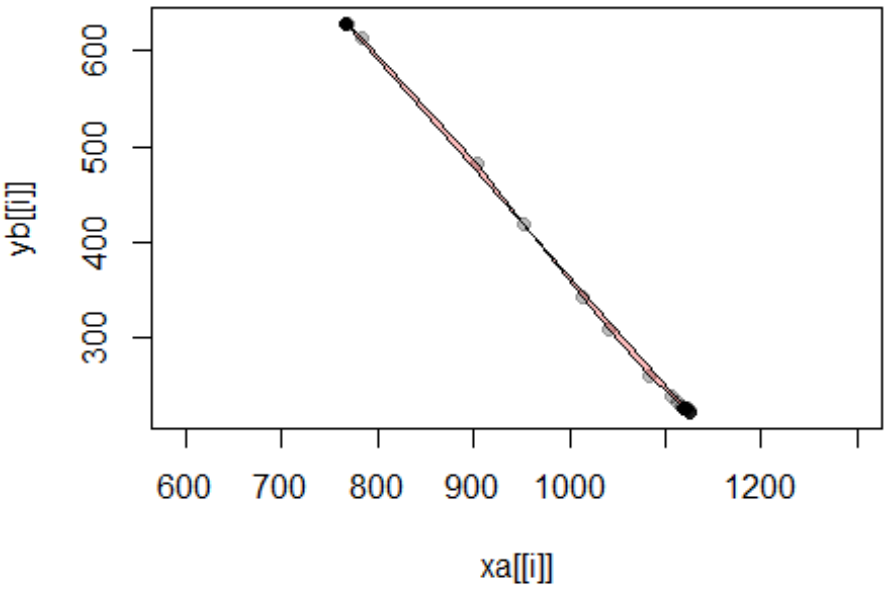

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

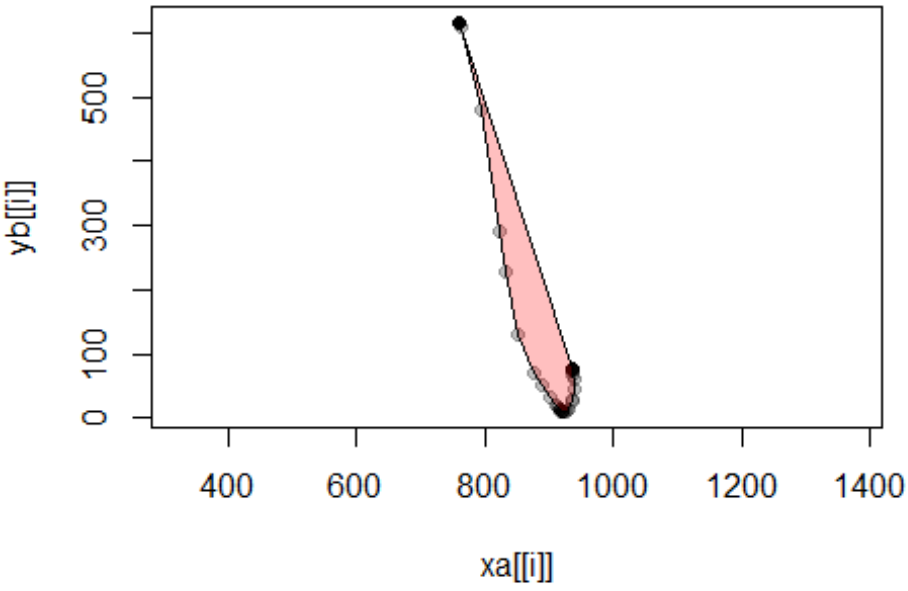

yb[[i]]
600
500
400
300
200
400
600
800
1000
xa[[i]]

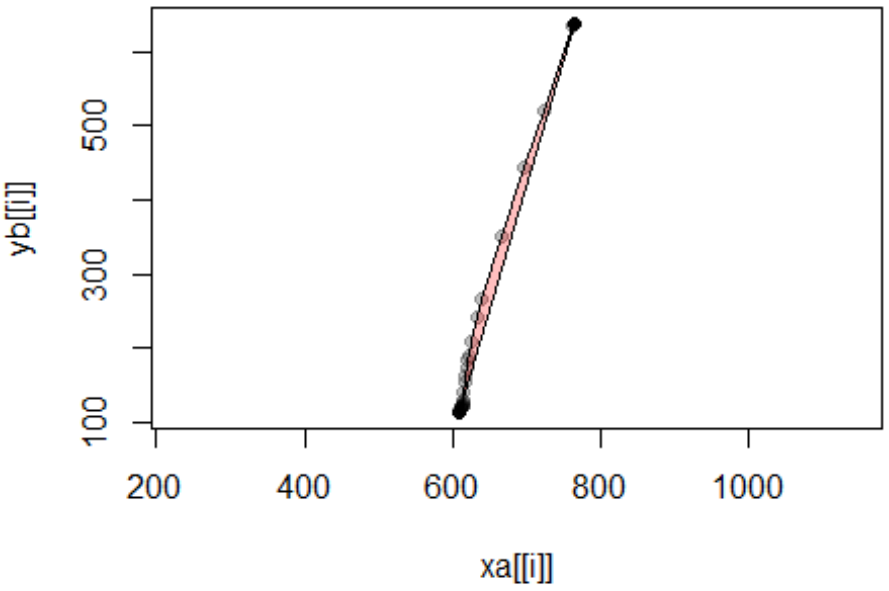

yb[[i]]
0
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

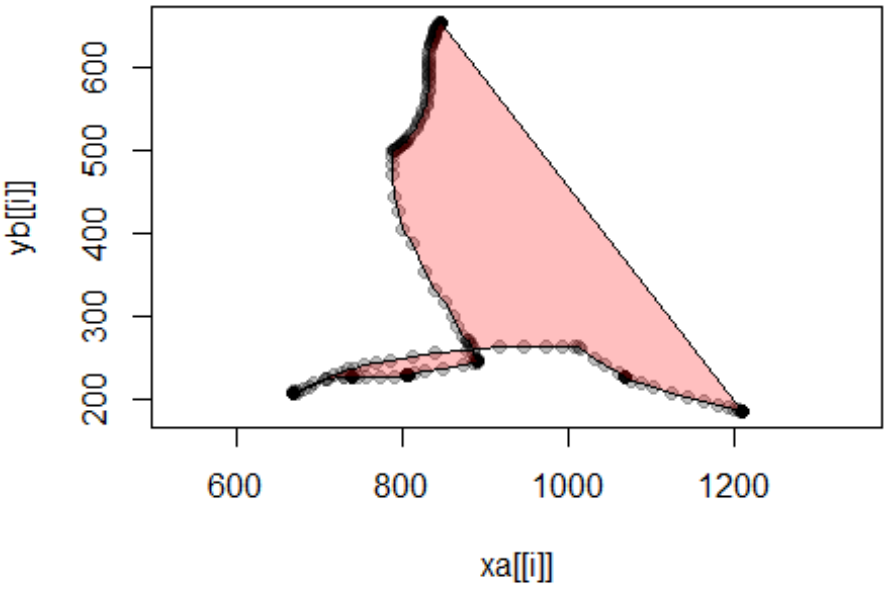

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

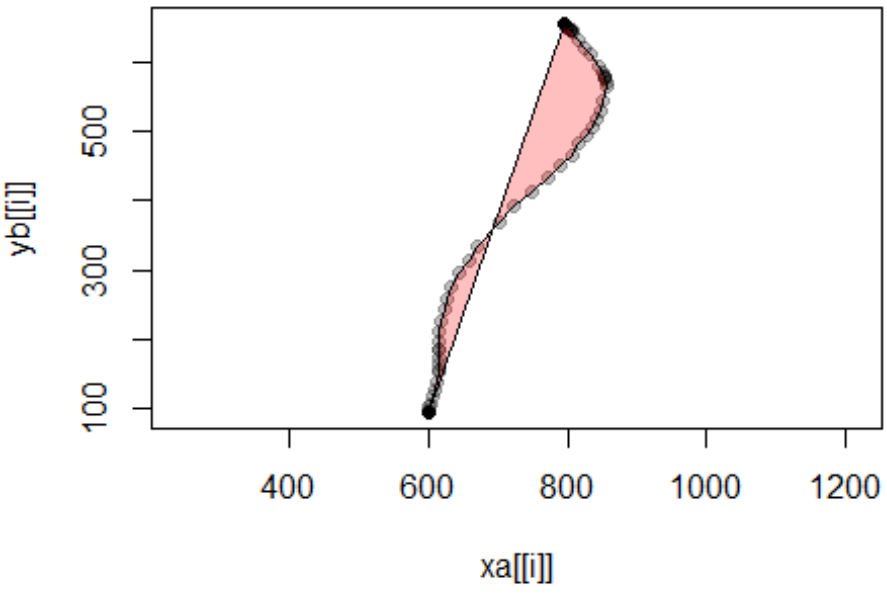

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

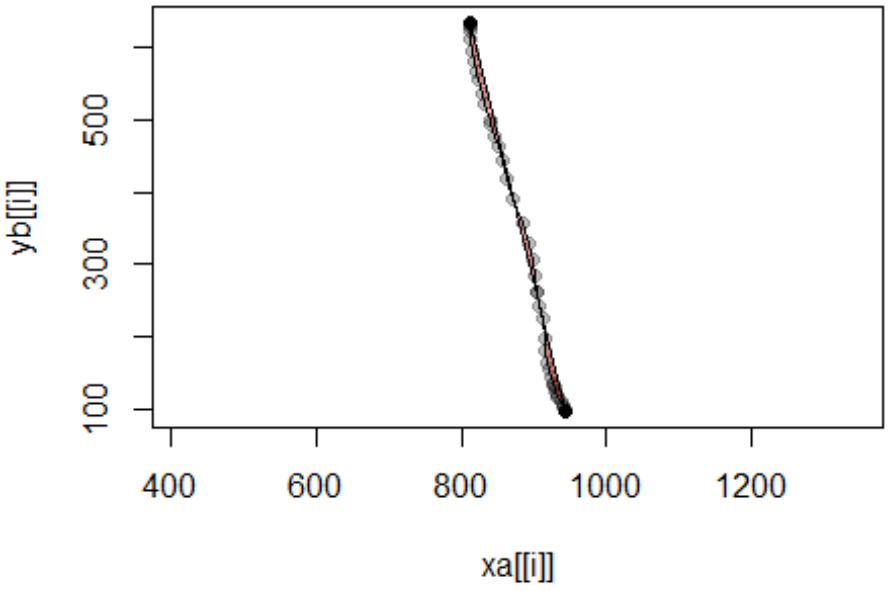

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

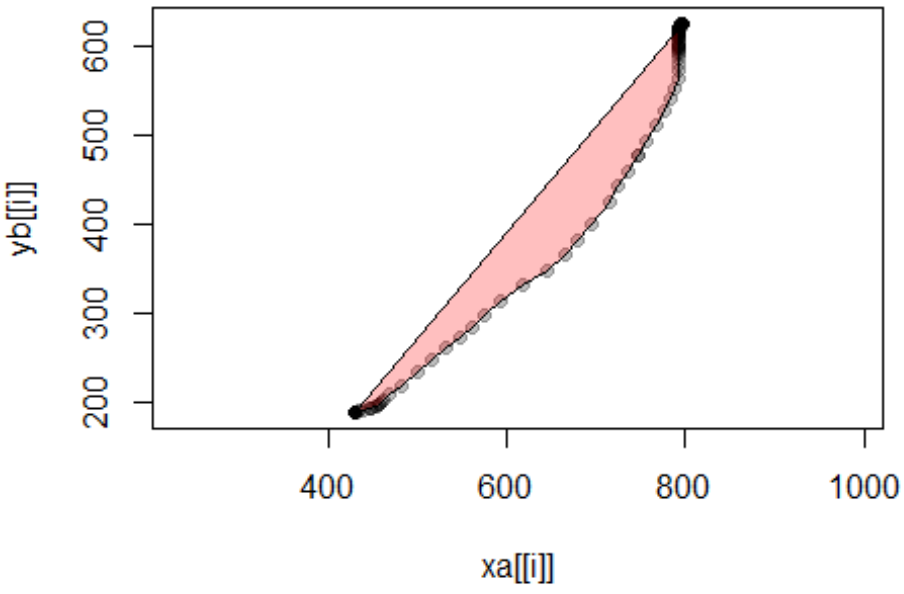

yb[[i]]
0
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

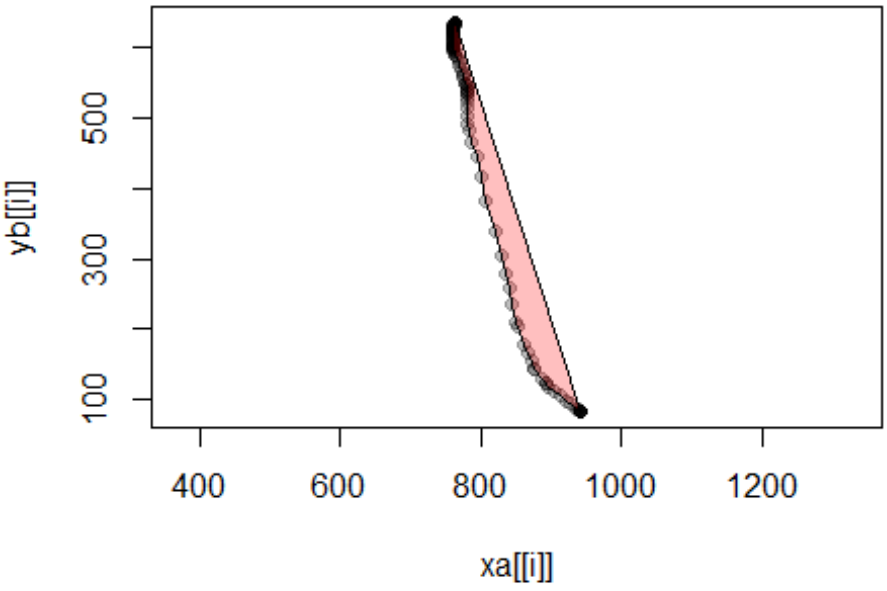

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

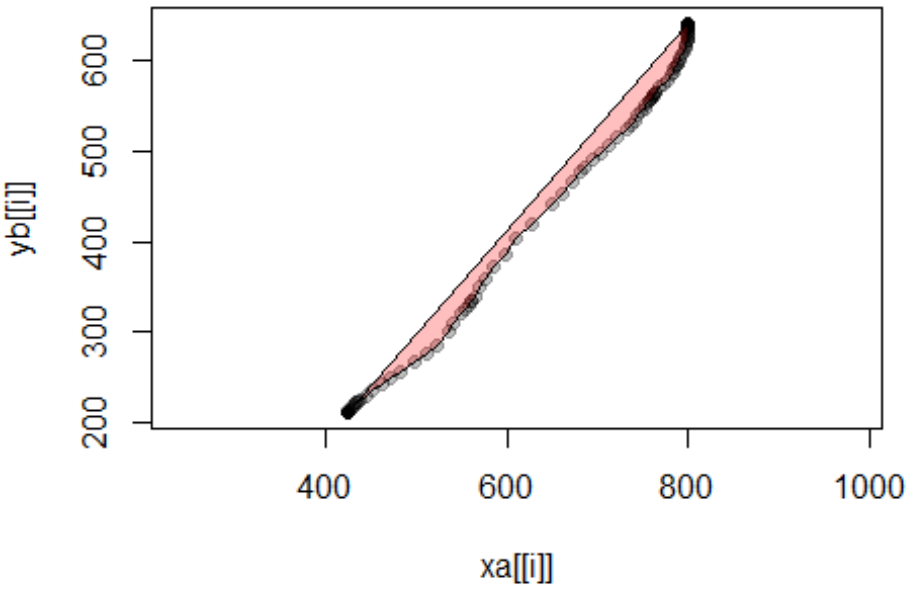

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

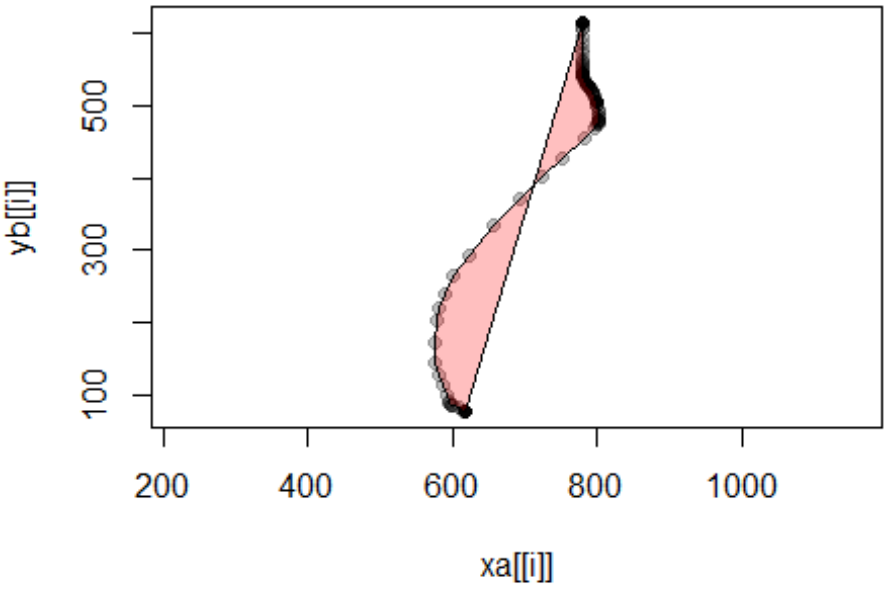

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

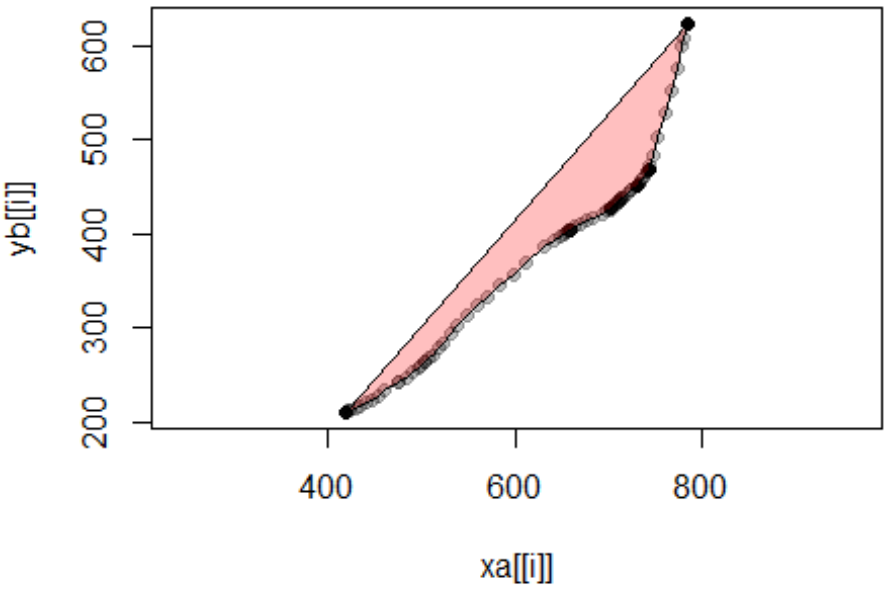

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

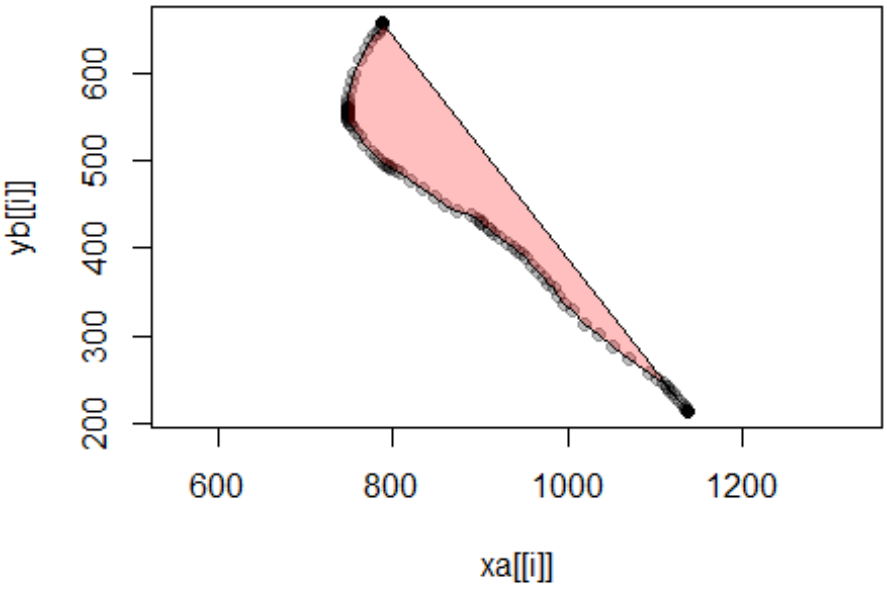

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

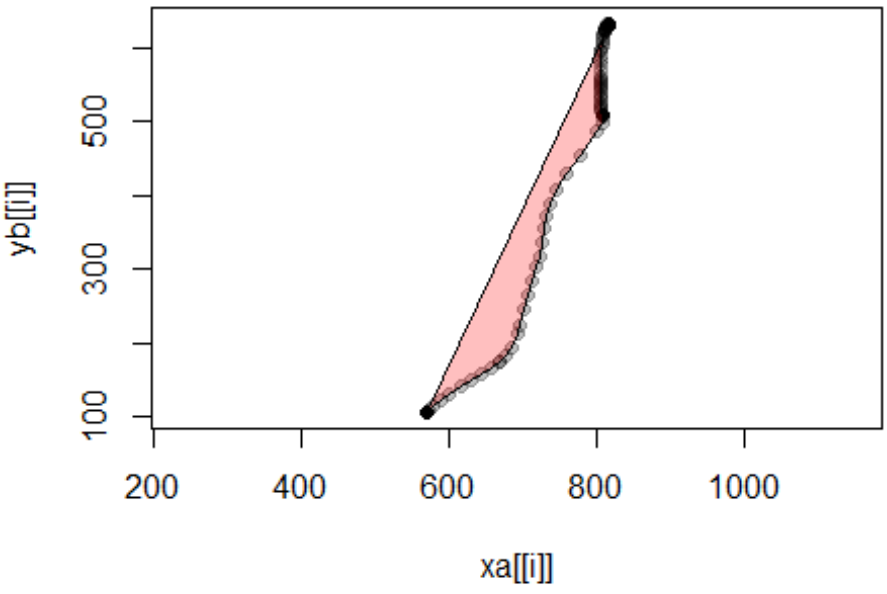

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

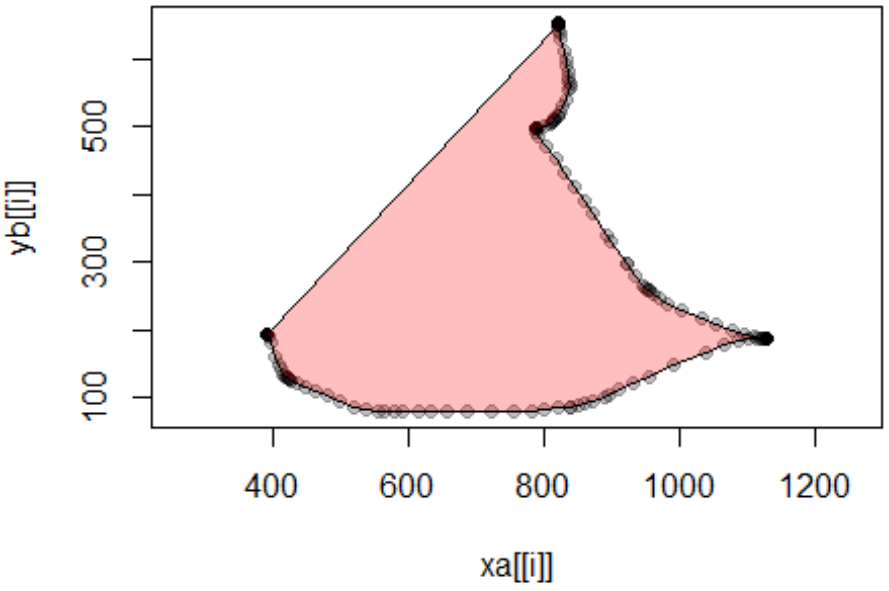

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

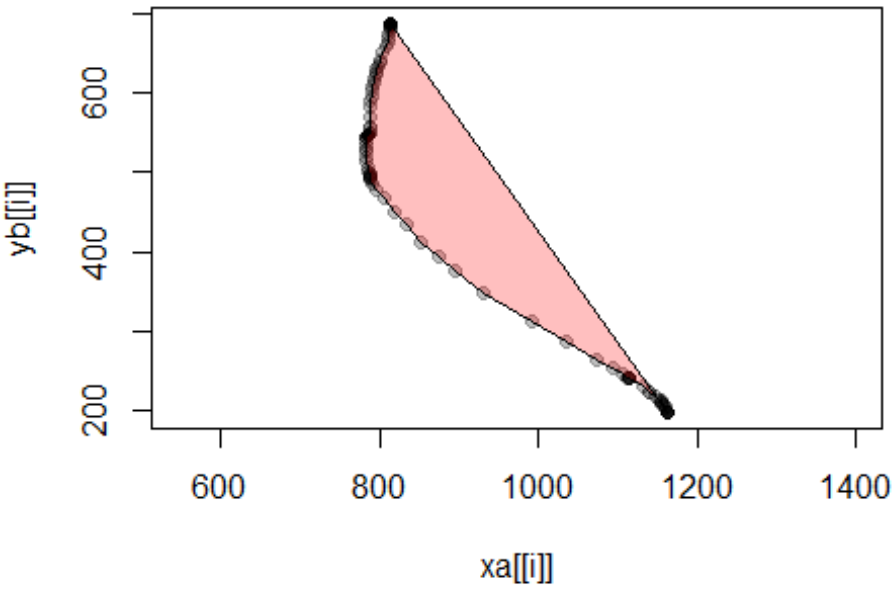

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

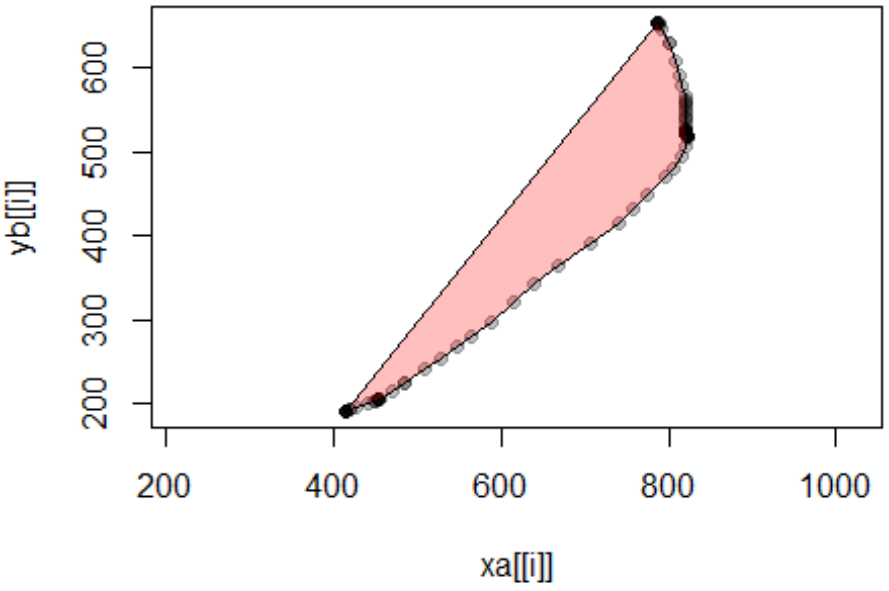

yb[[i]]
200
300
400
500
600
400
600
800
xa[[i]]

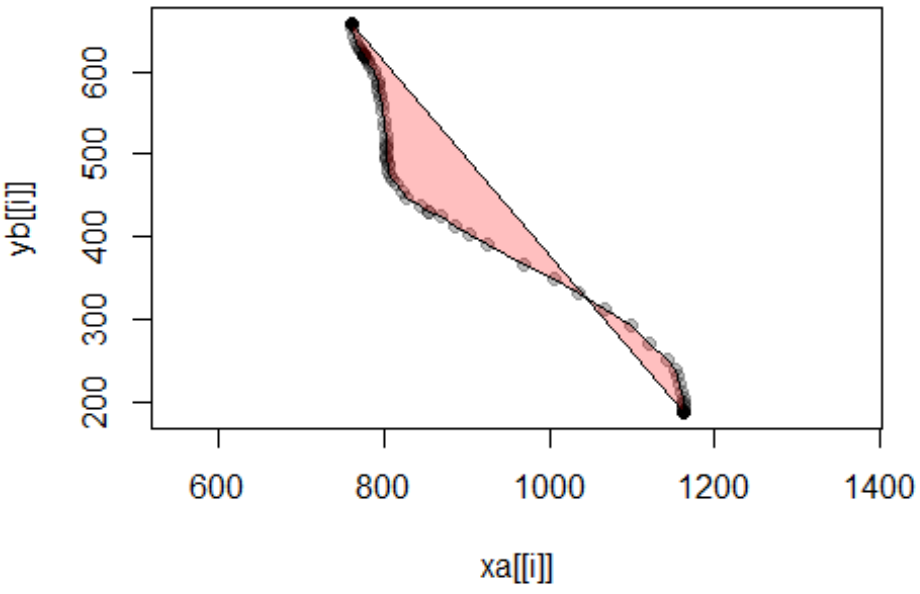

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

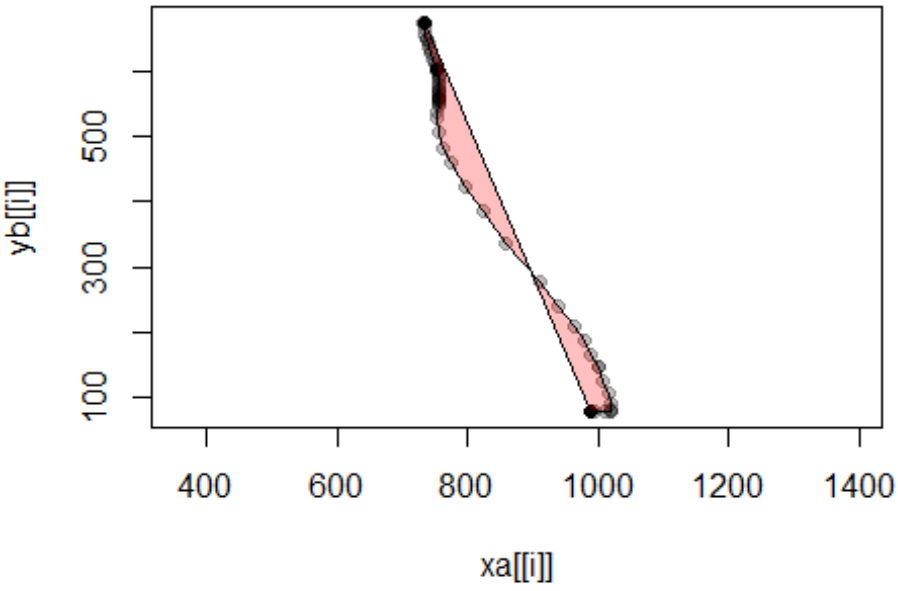

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

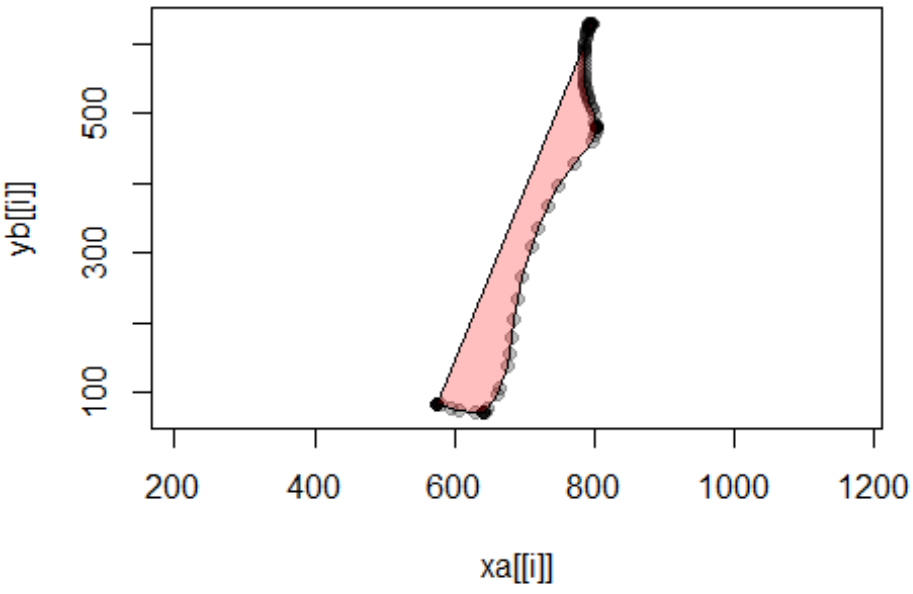

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

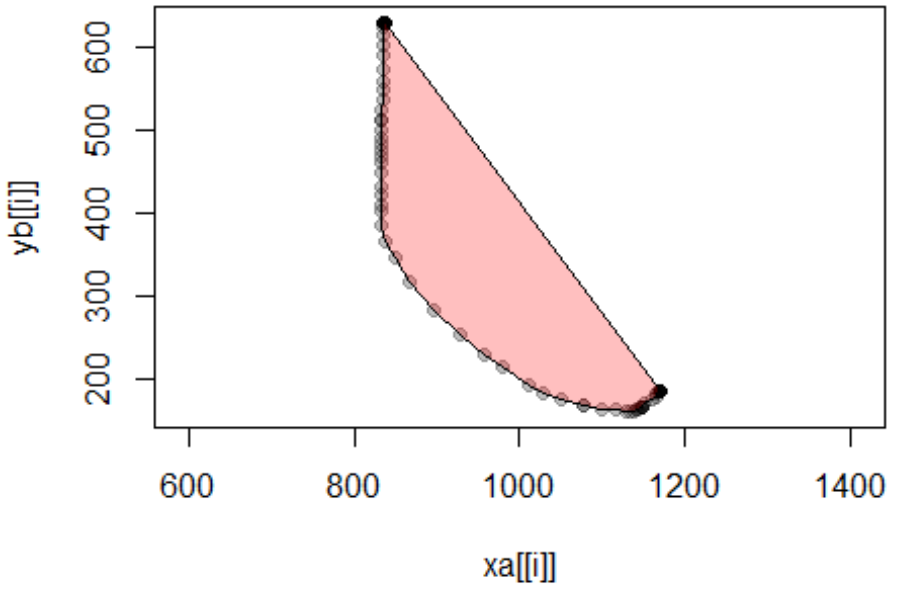

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

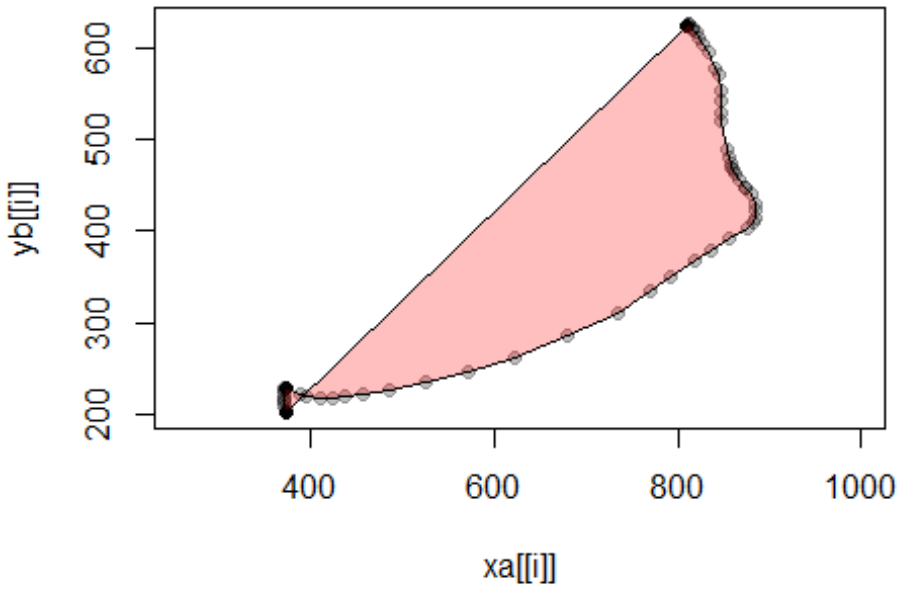

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

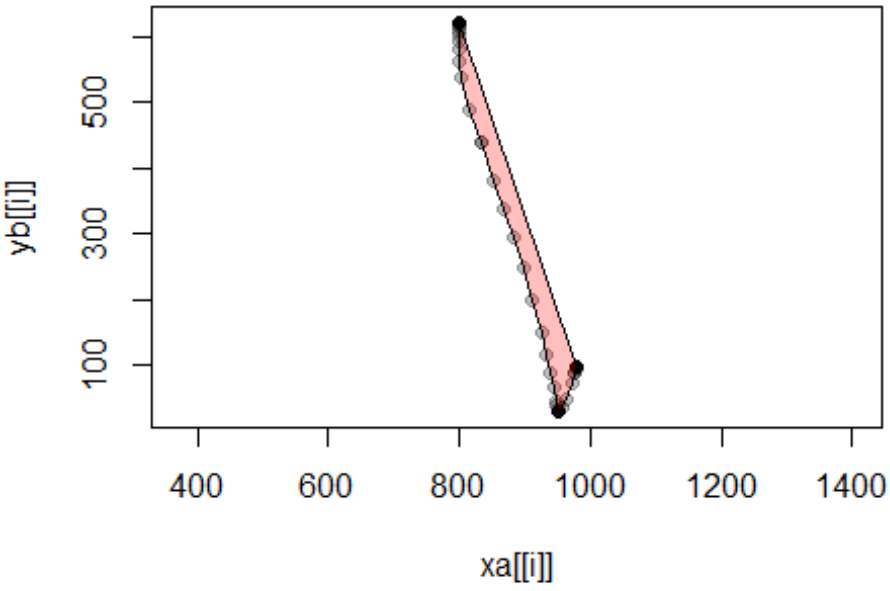

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

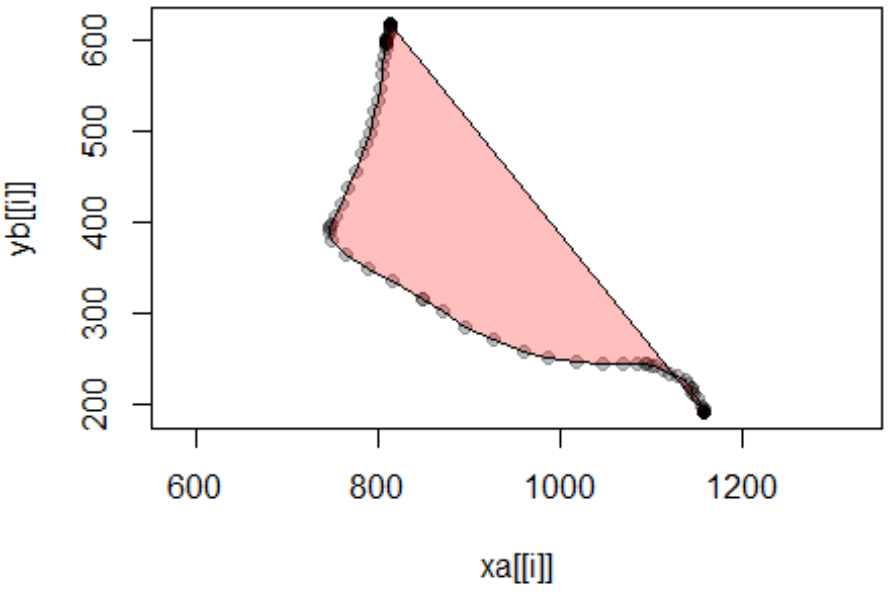

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

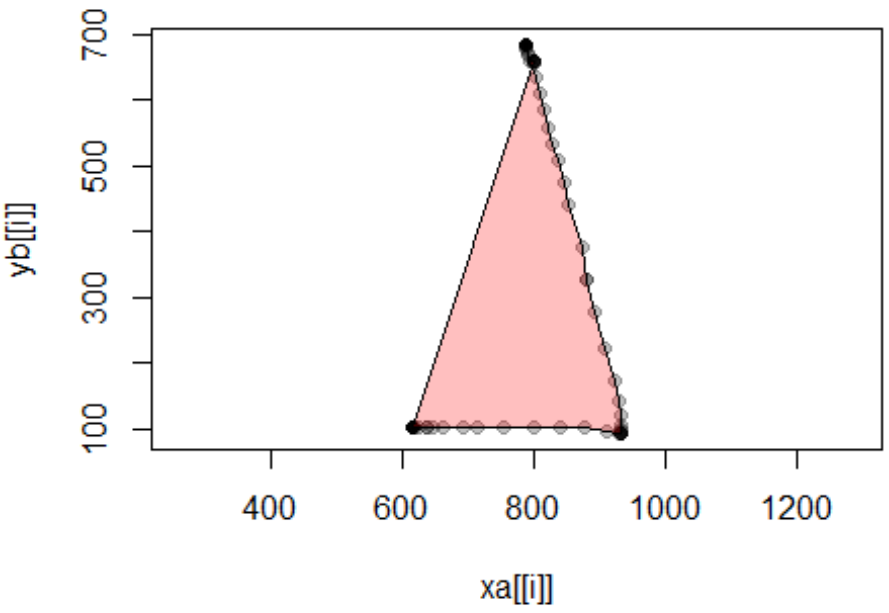

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

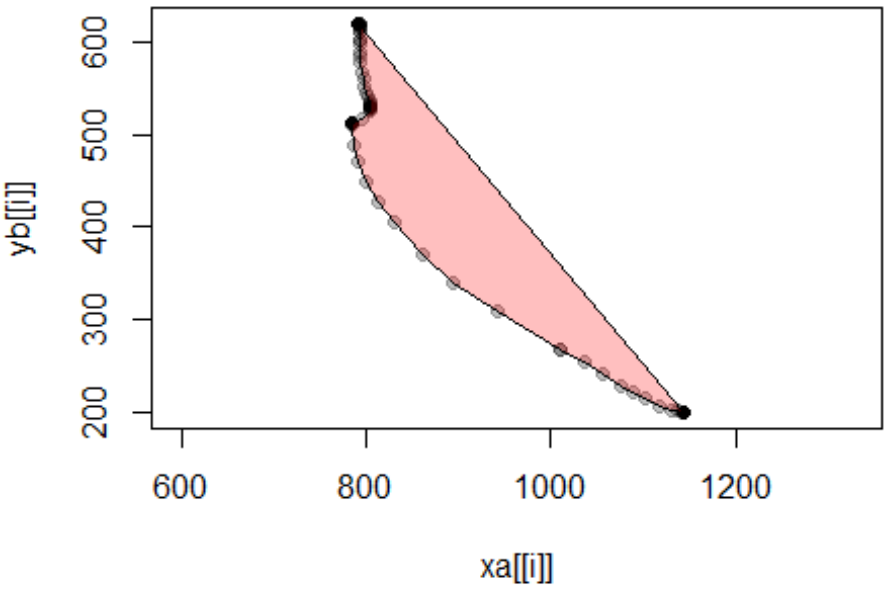

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

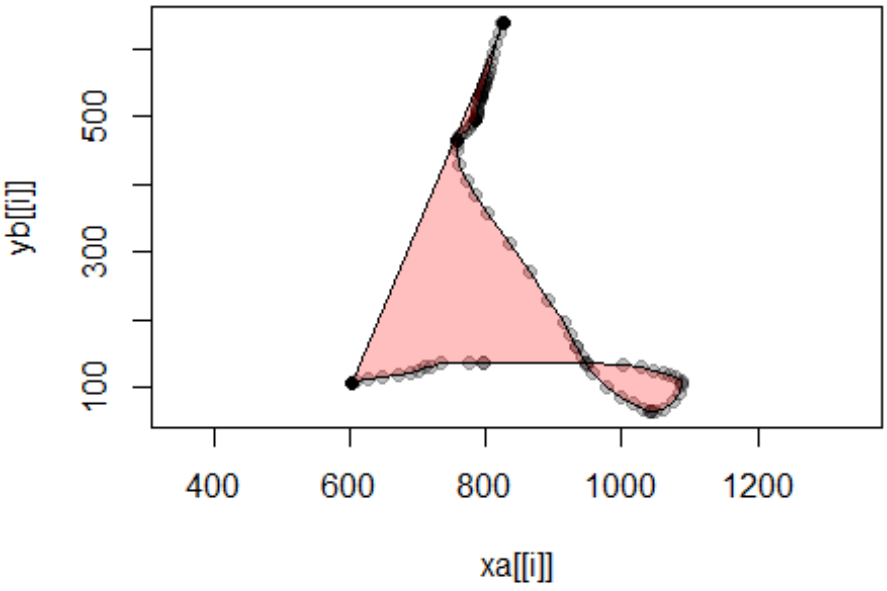

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

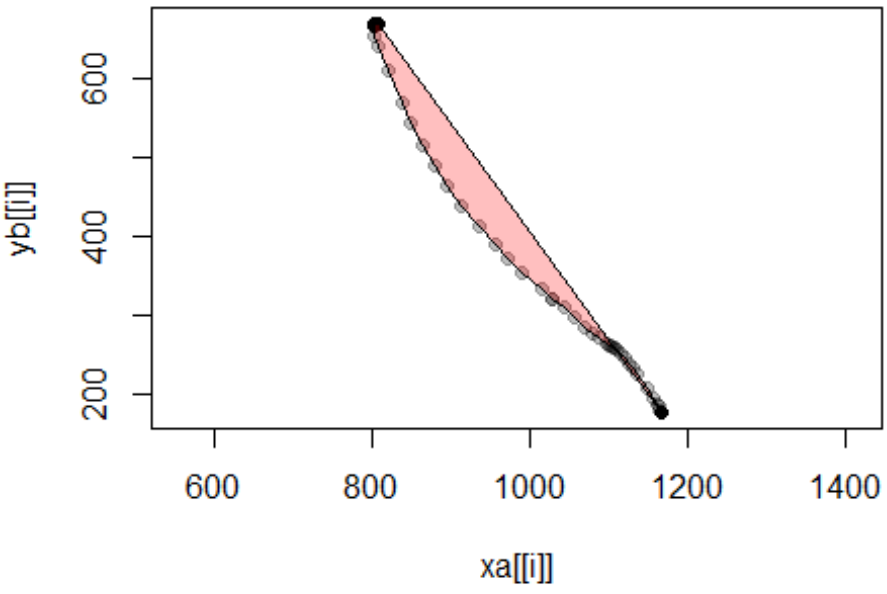

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

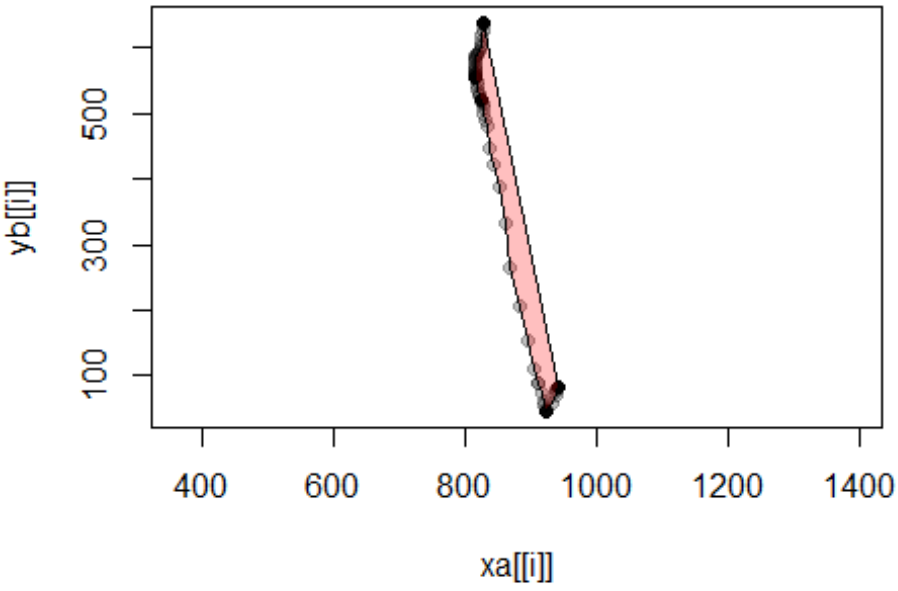

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

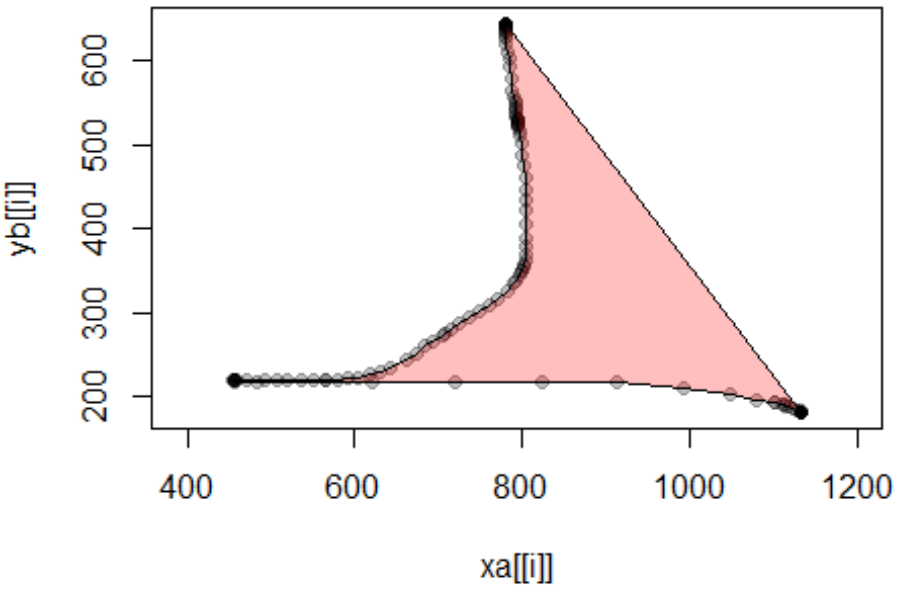

yb[[i]]
100
300
500
700
400
600
800
1000
1200
xa[[i]]

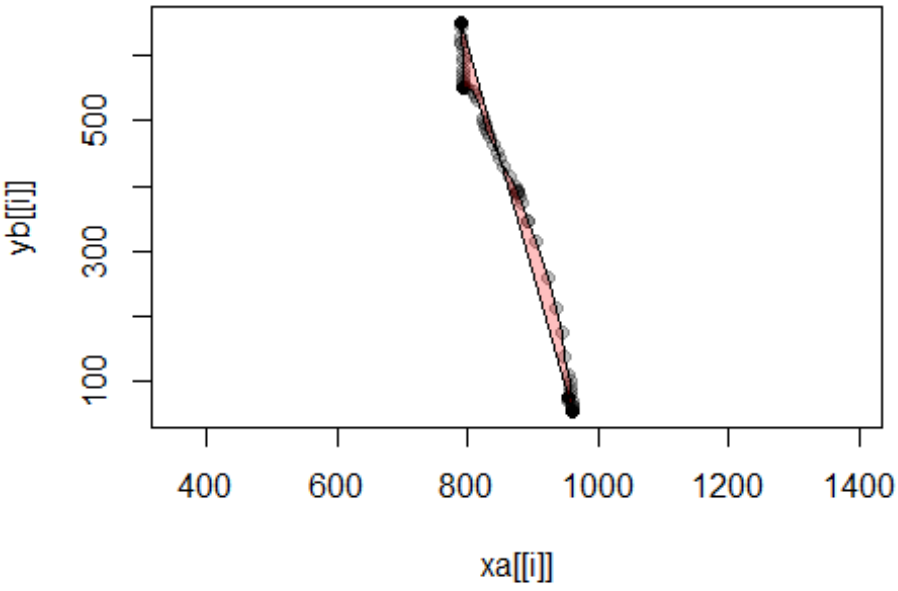

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

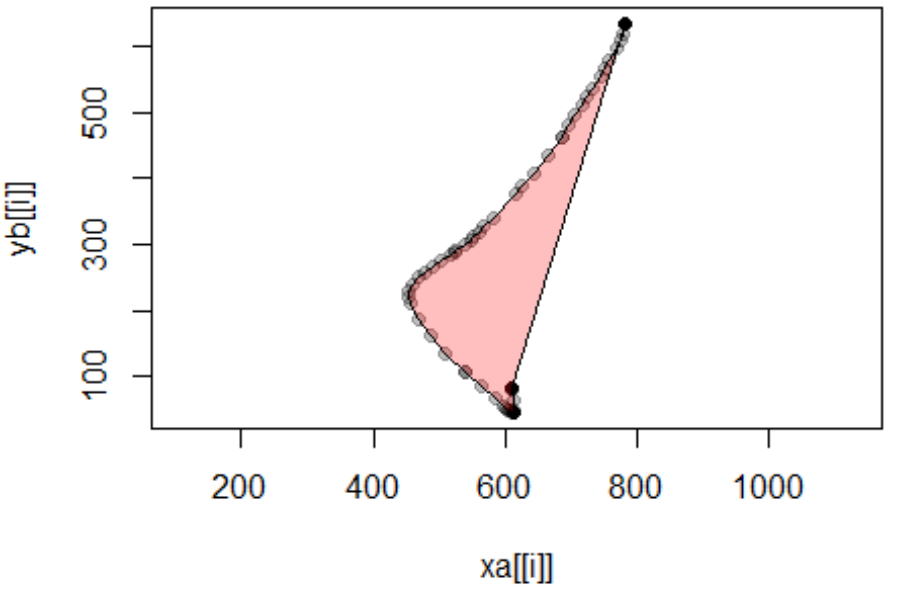

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

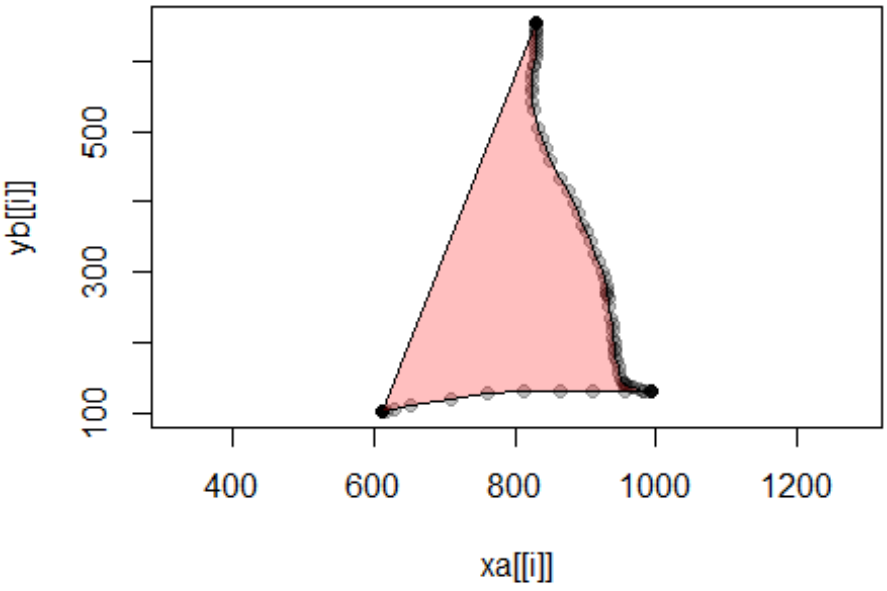

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

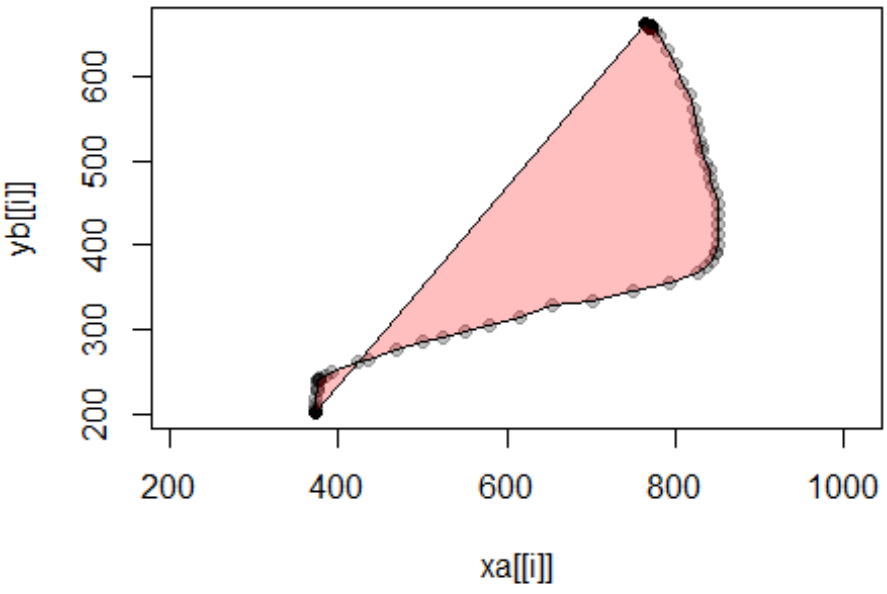

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

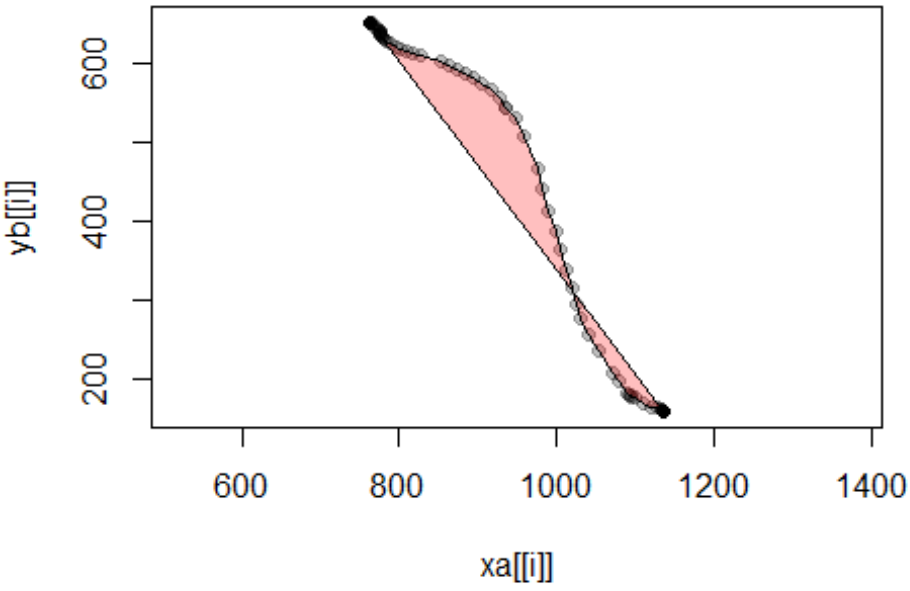

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

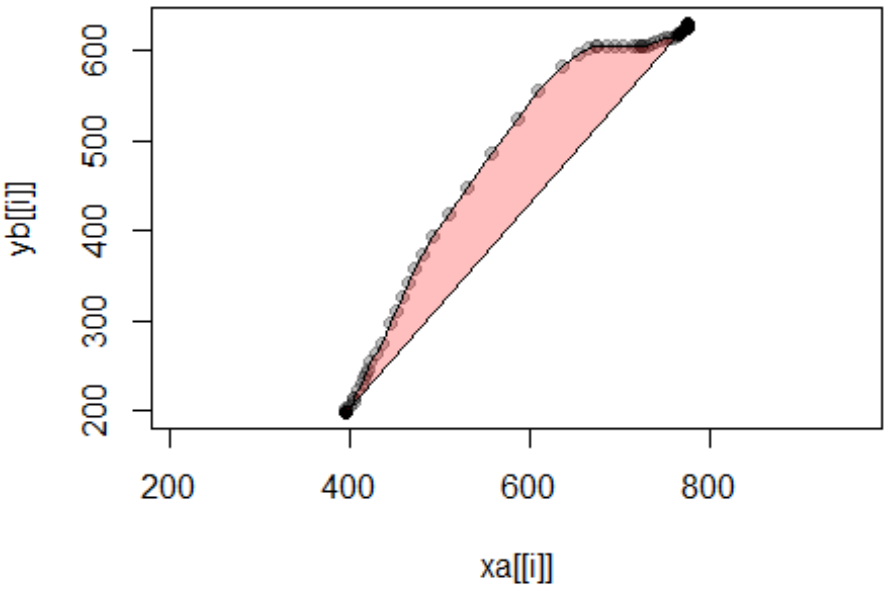

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

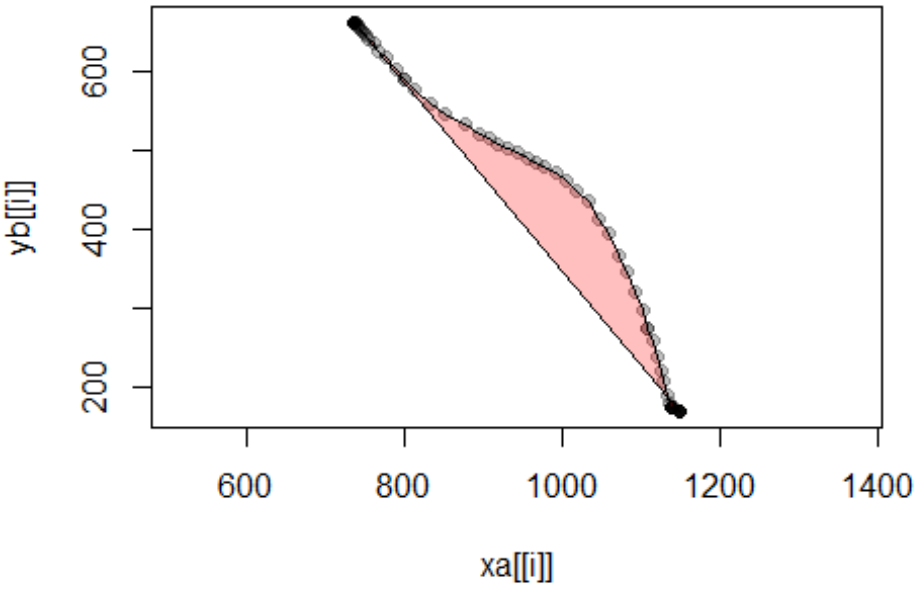

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

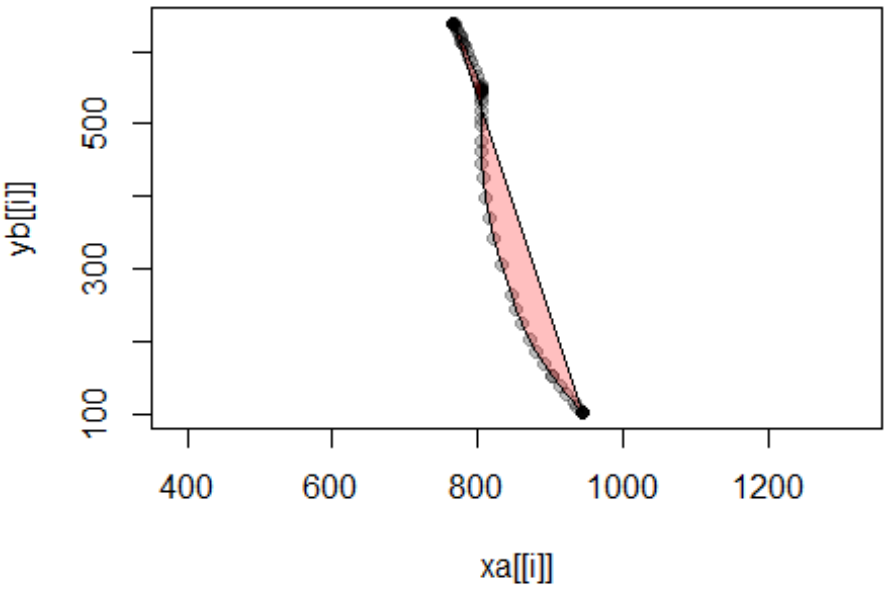

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

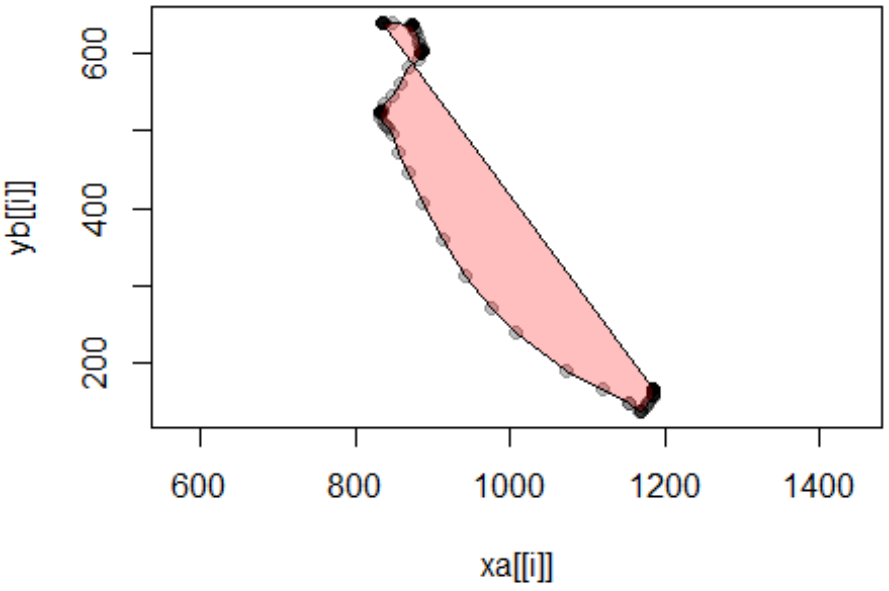

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

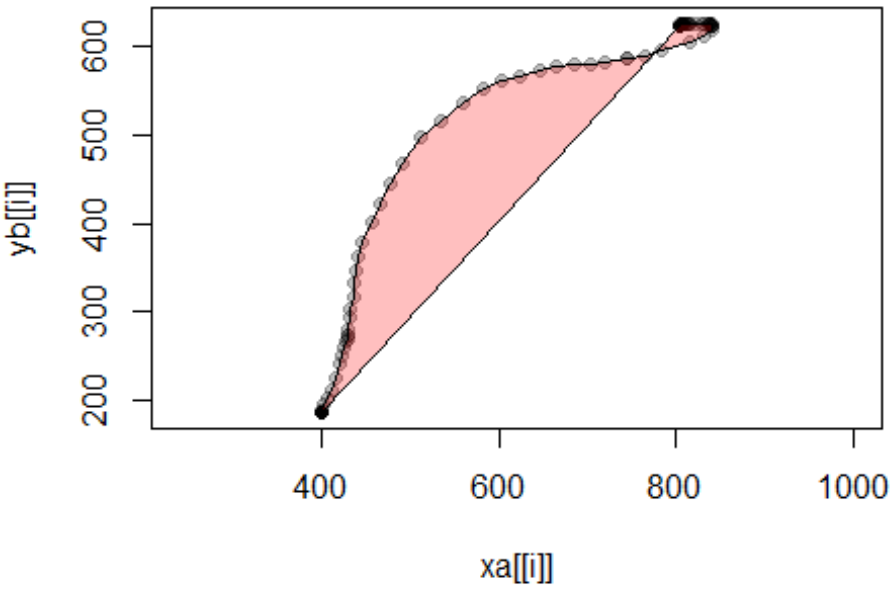

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

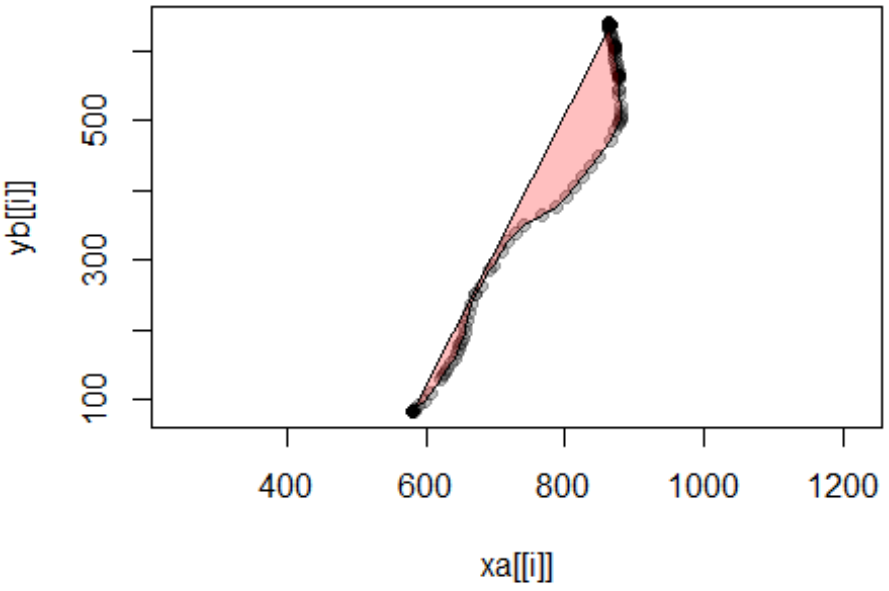

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

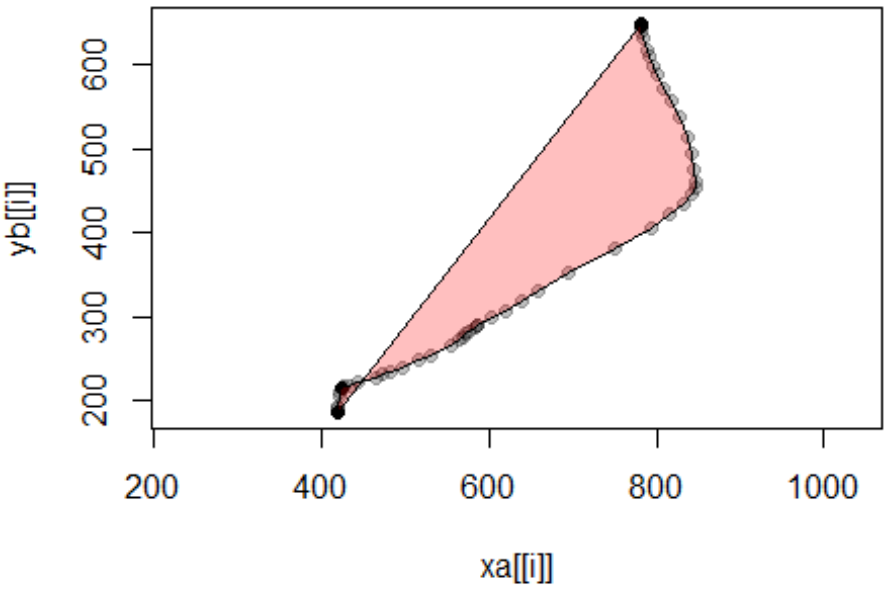

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

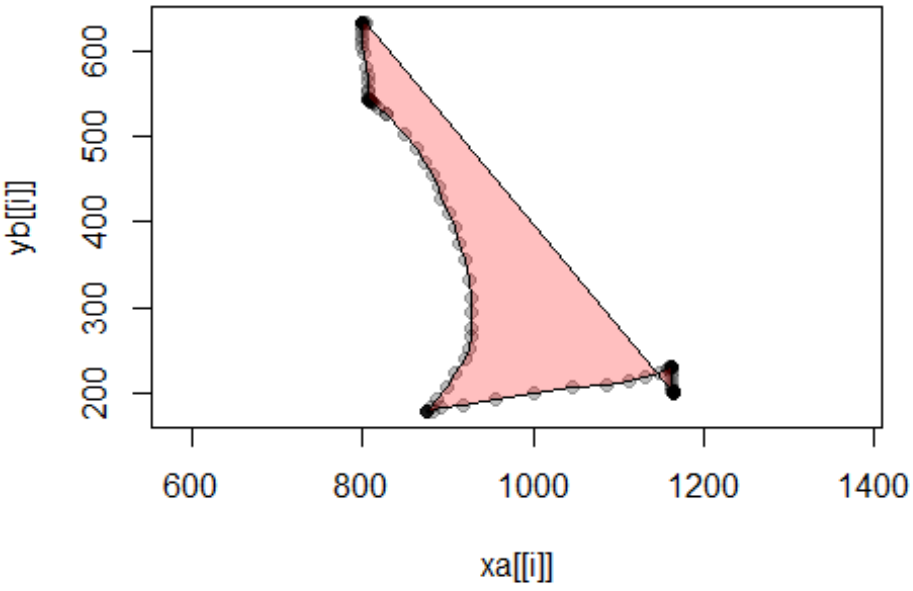

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

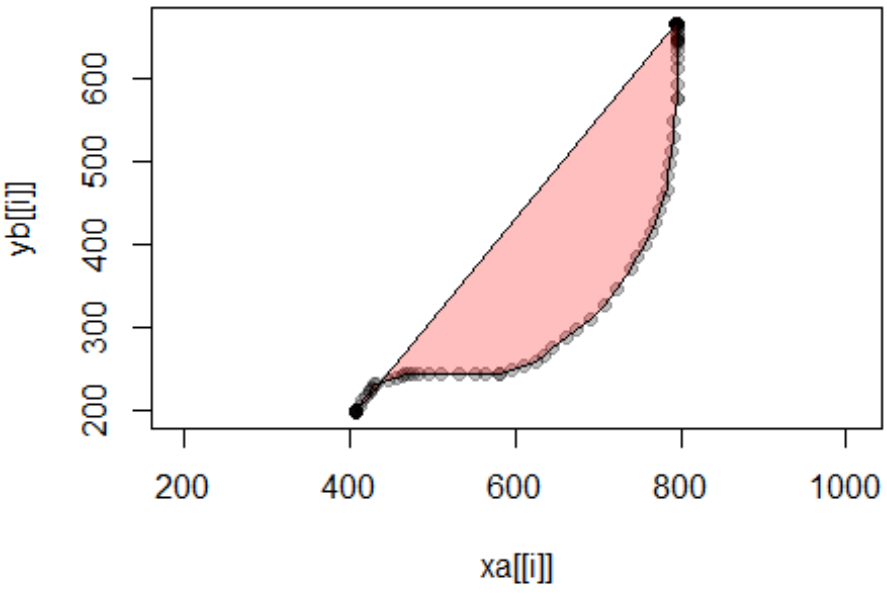

yb[[i]]
600
400
200
600
800
1000
1200
1400
xa[[i]]

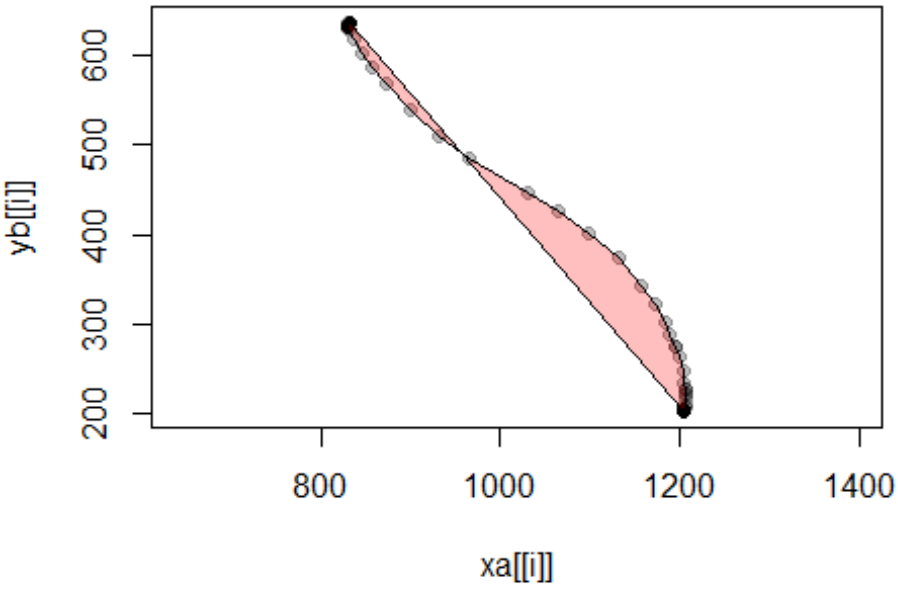

yb[[i]]
600
500
400
300
200
400
600
800
1000
xa[[i]]

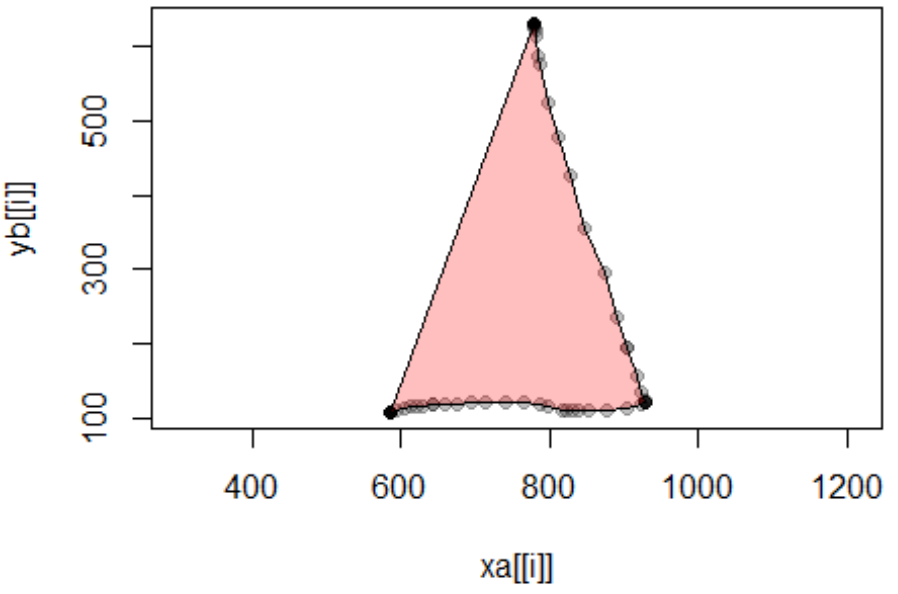

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

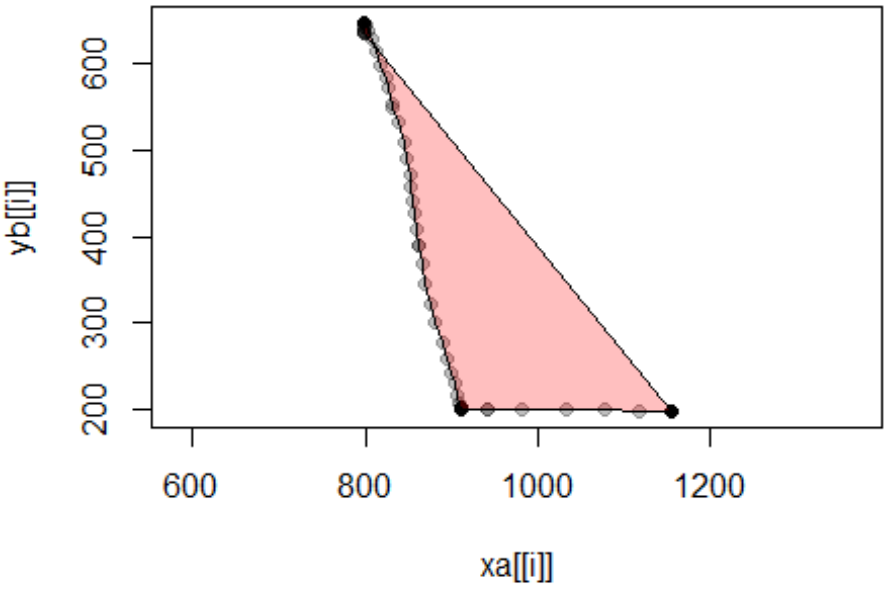

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

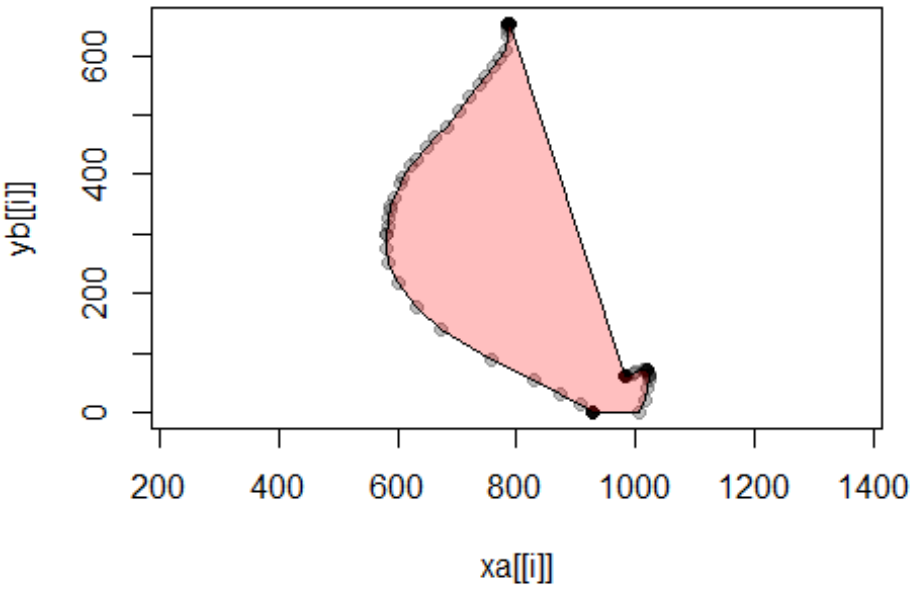

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

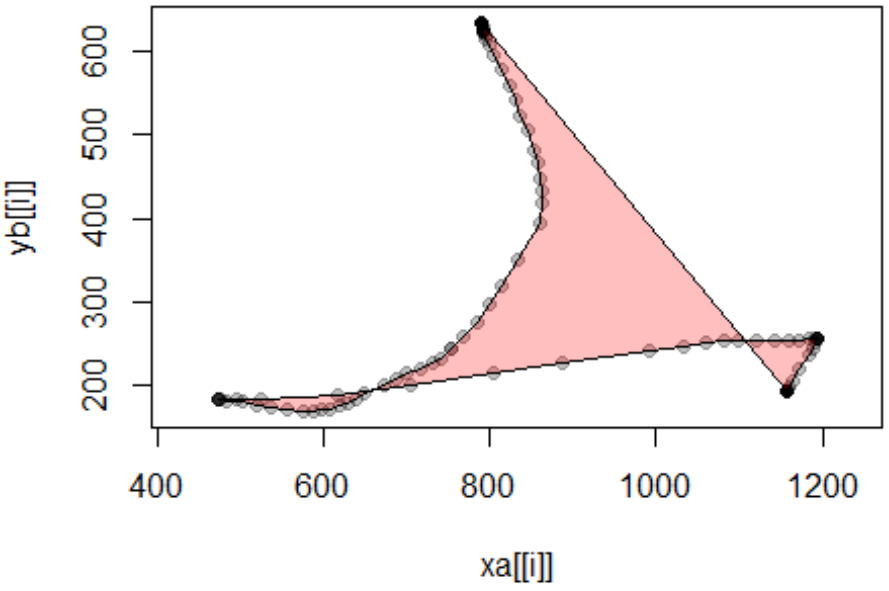

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

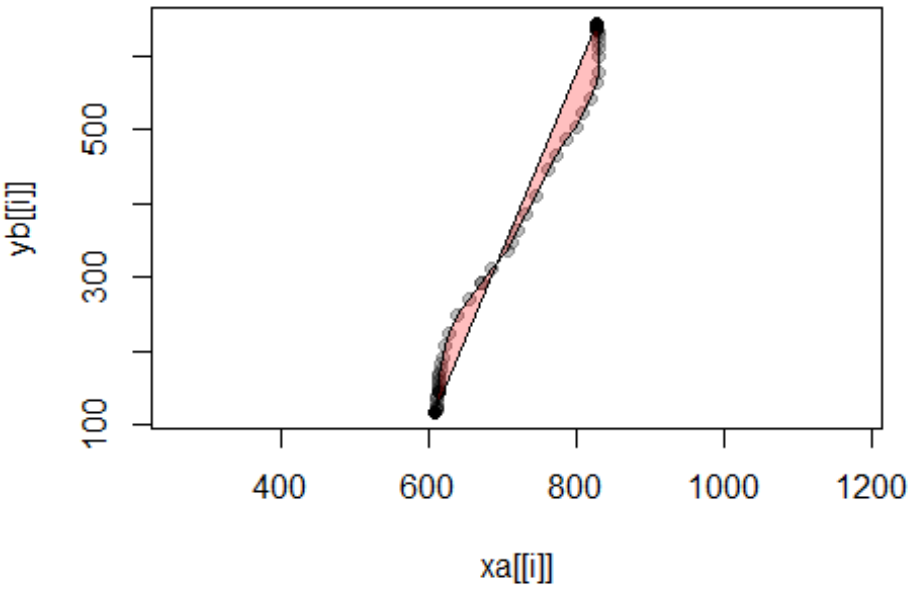

yb[[i]]
200
300
400
500
600
800
1000
1200
1400
xa[[i]]

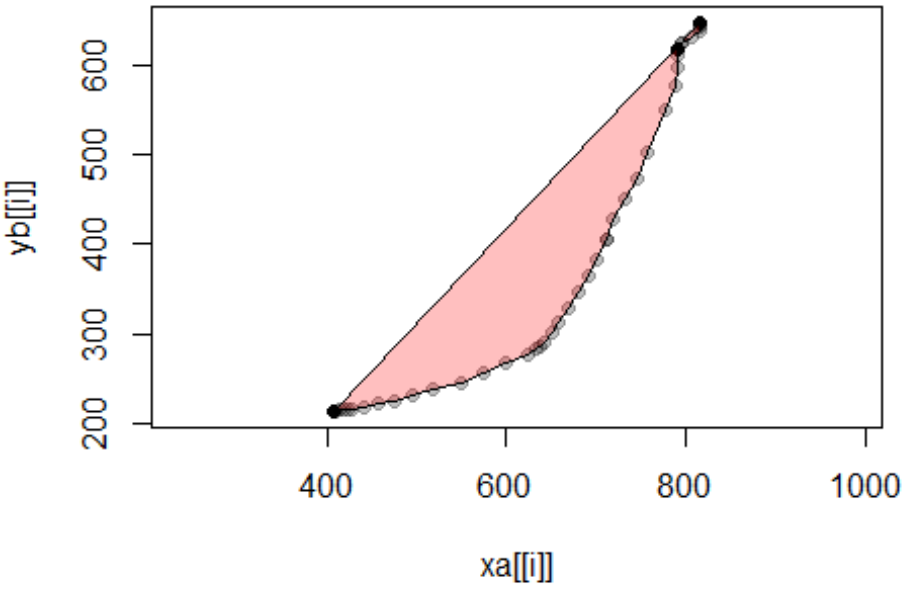

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

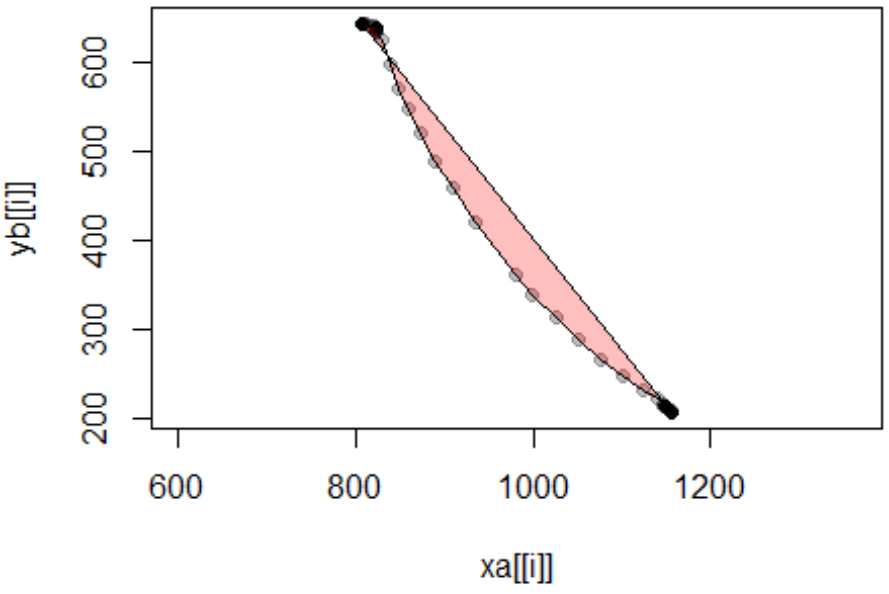

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

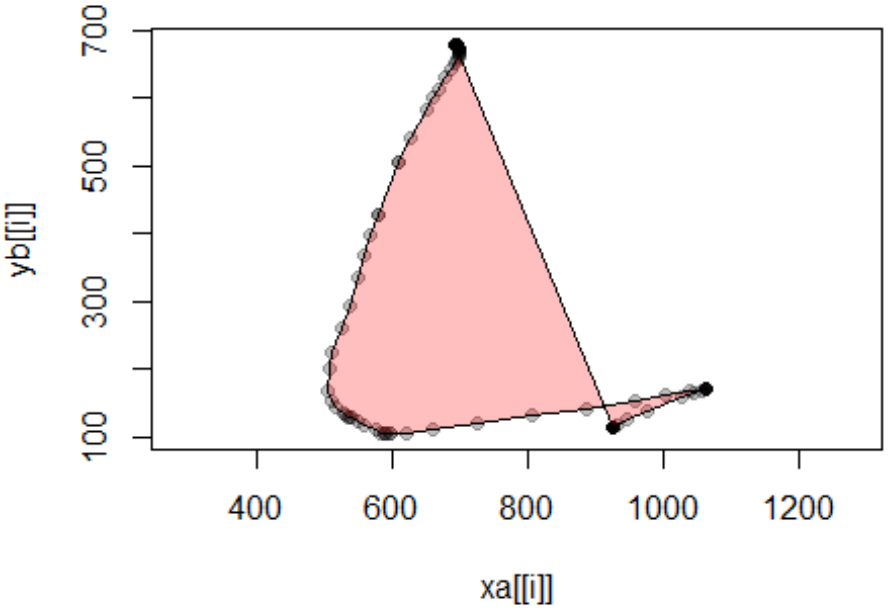

yb[[i]]
0
200
400
600
200
400
600
800
1000
1200
1400
xa[[i]]

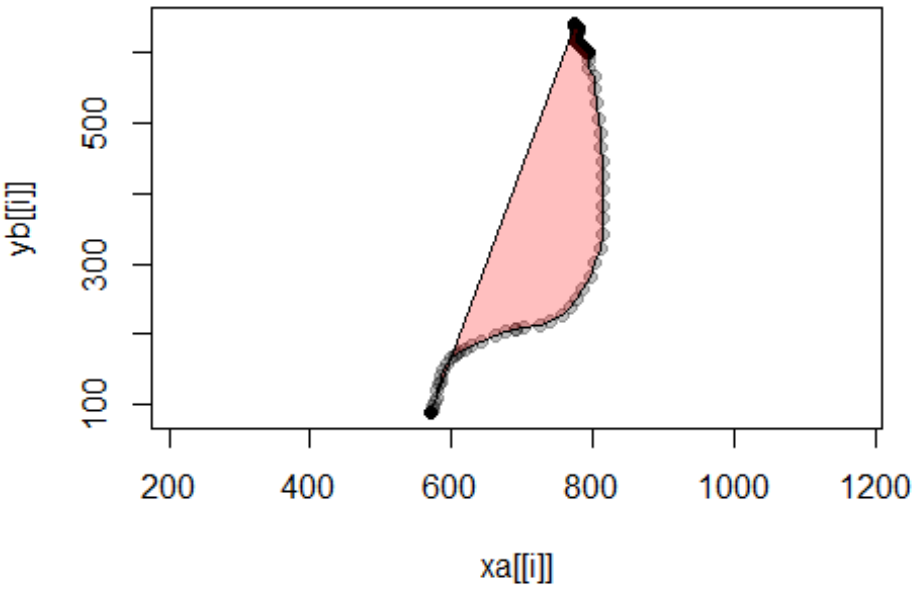

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

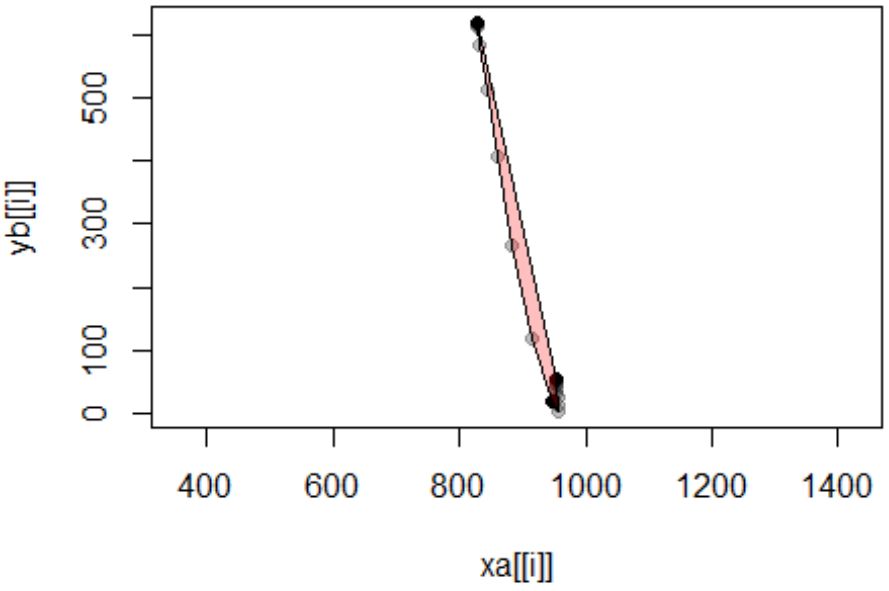

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

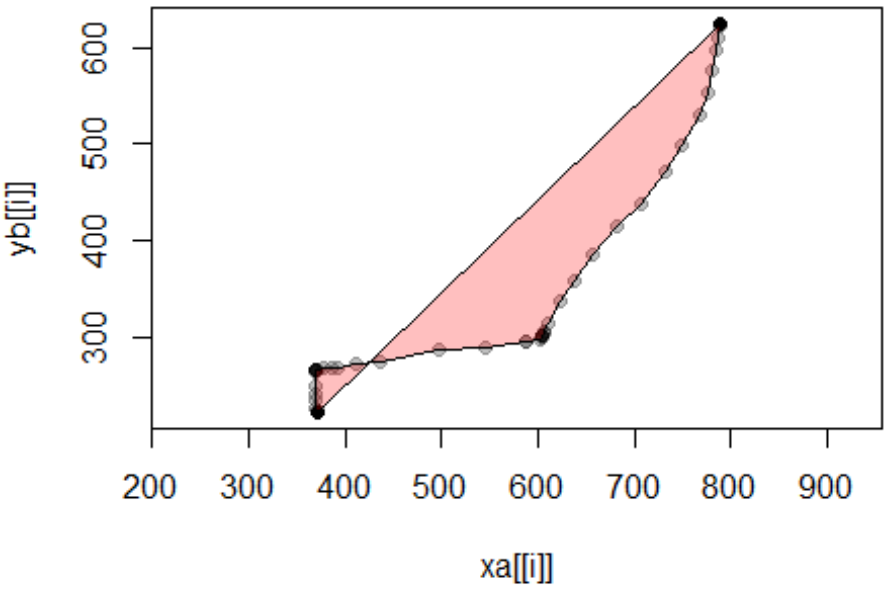

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

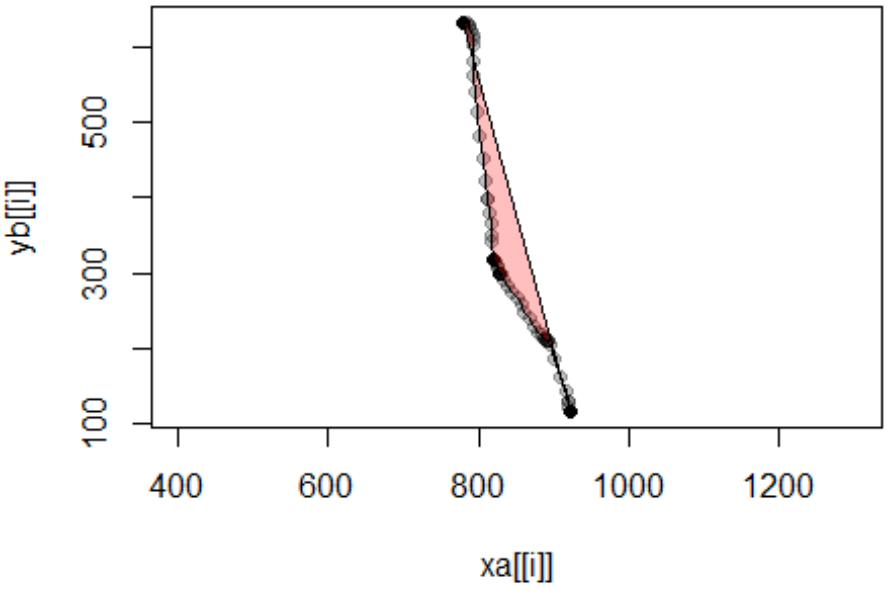

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

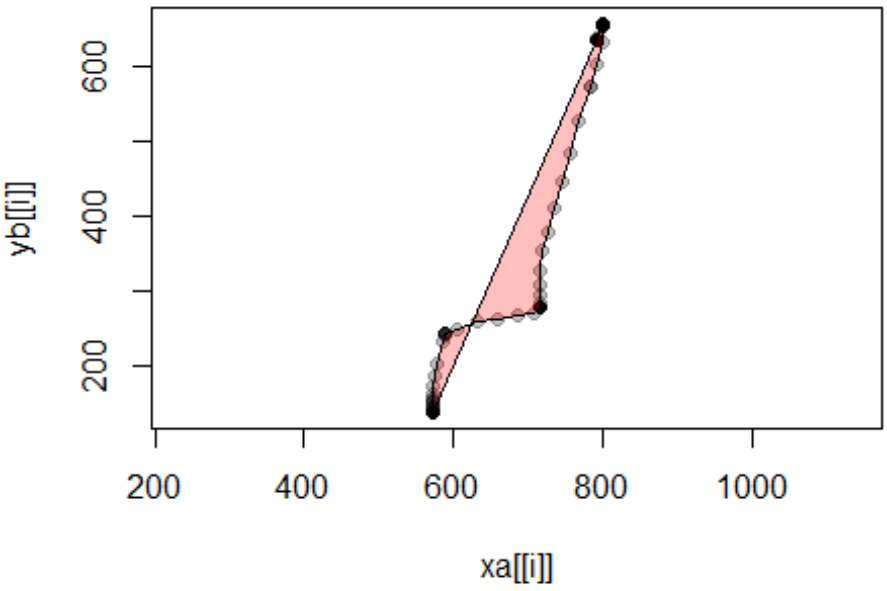

yb[[i]]
100
300
500
700
400
600
800
1000
1200
xa[[i]]

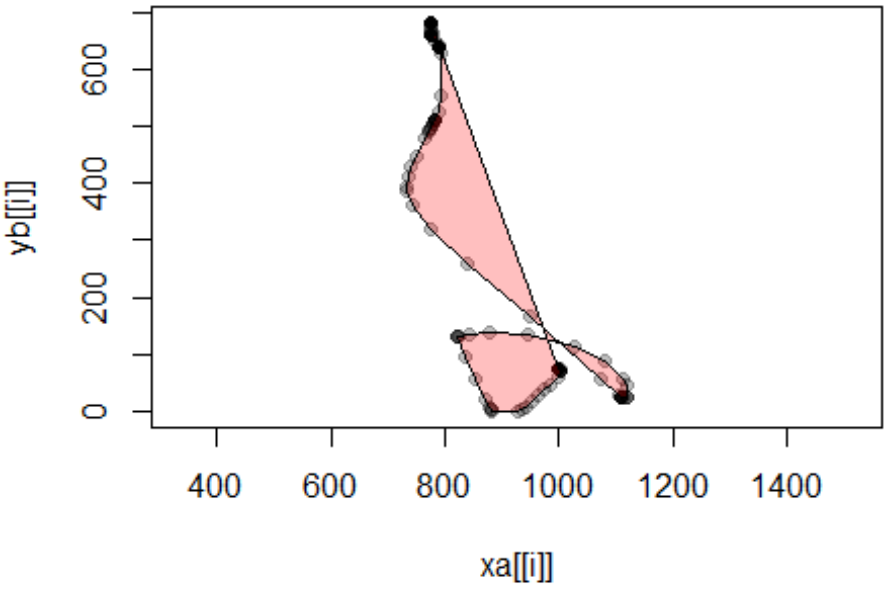

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

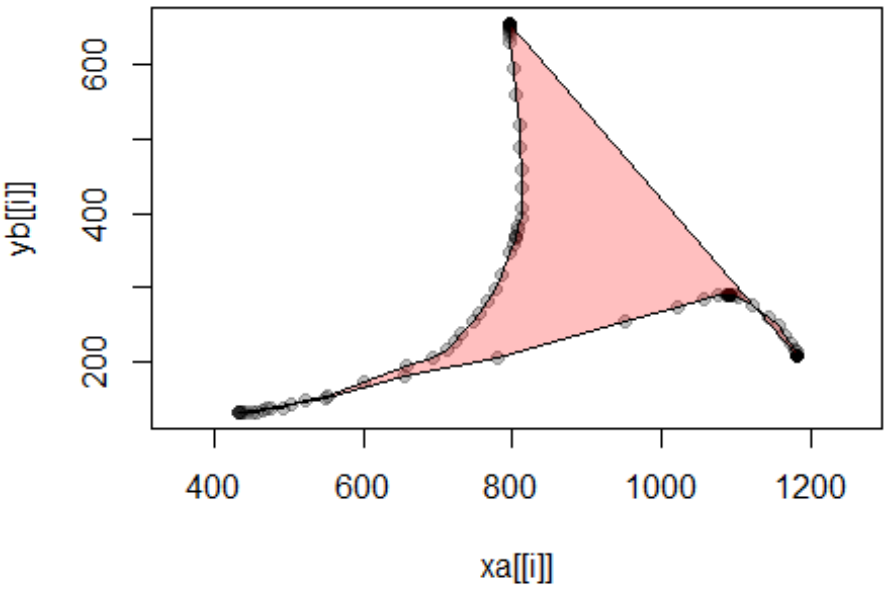

yb[[i]]
0
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

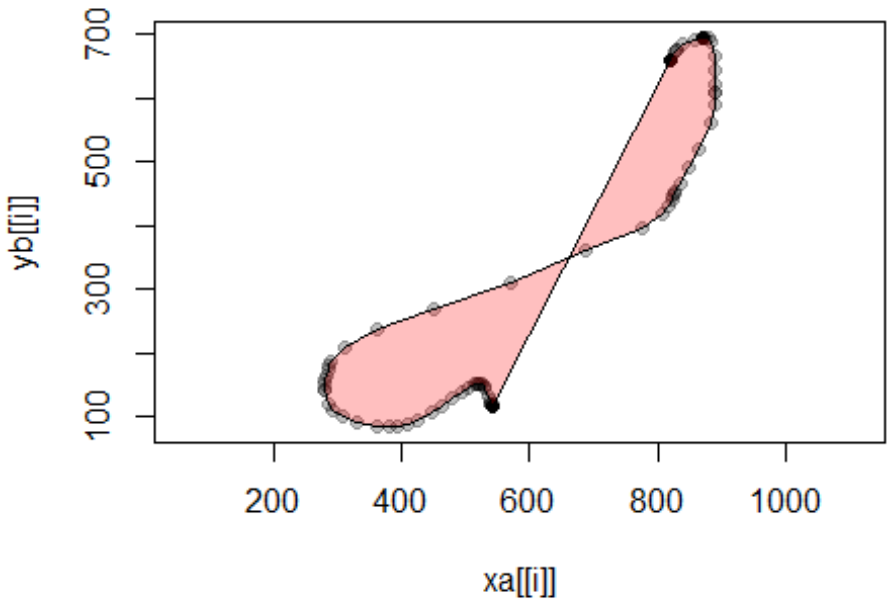

yb[[i]]
300
400
500
600
200
300
400
500
600
700
800
900
xa[[i]]

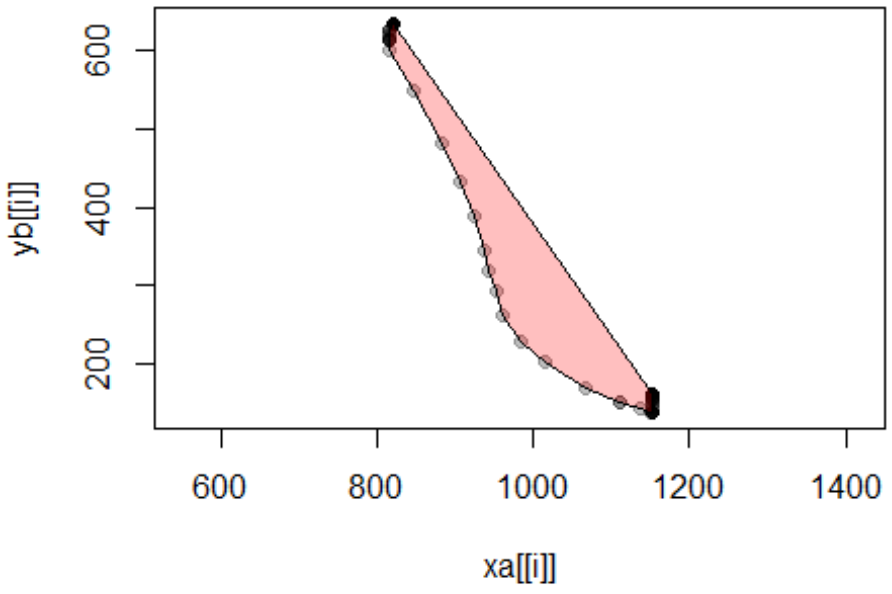

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

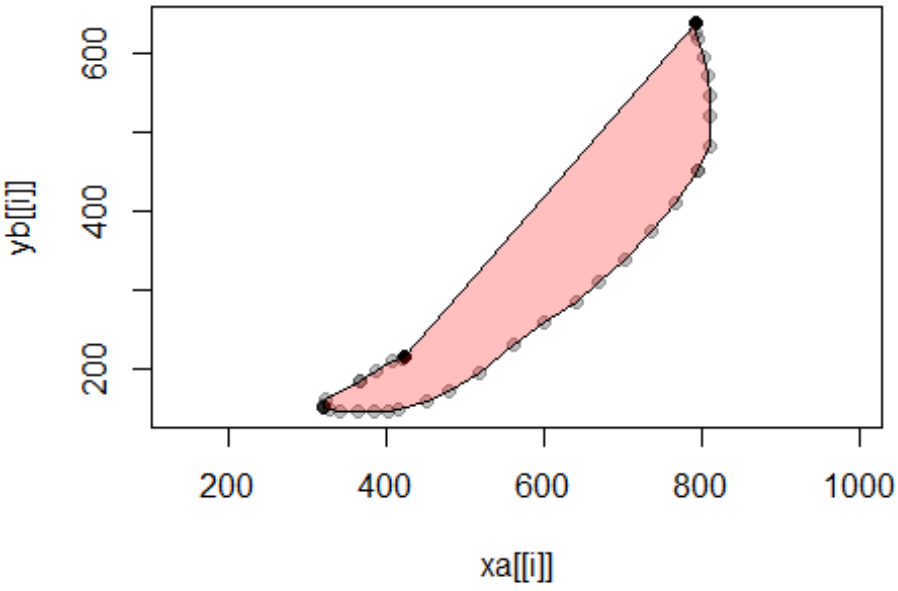

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

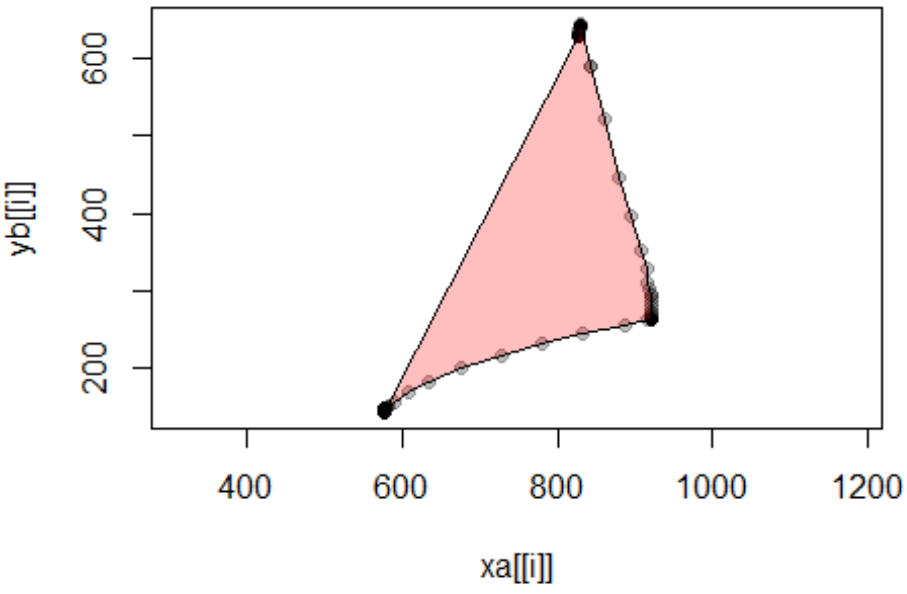

yb[[i]]
0
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

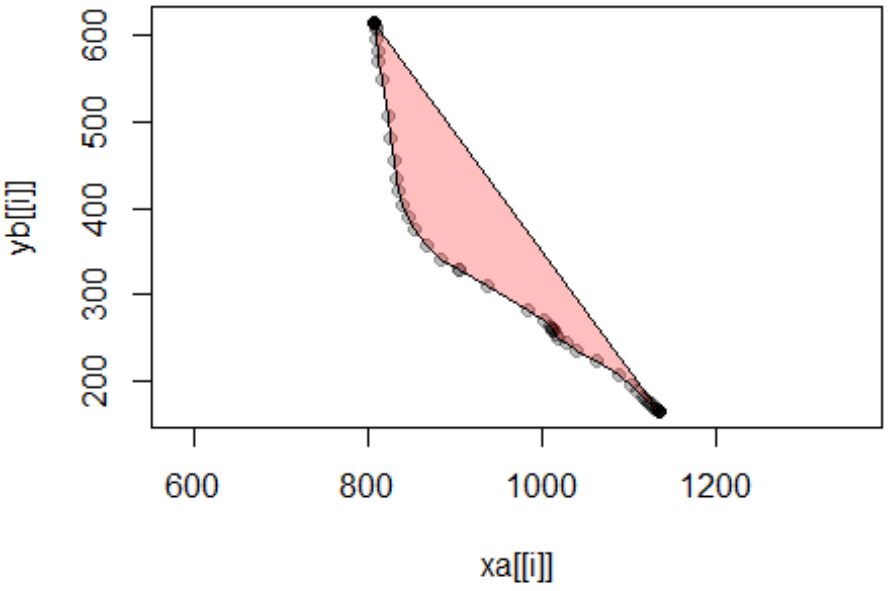

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

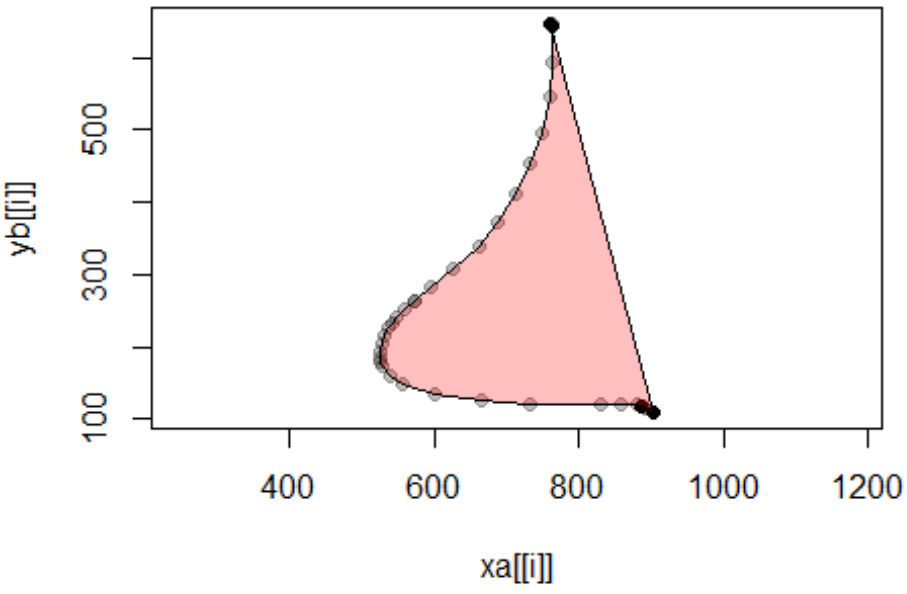

yb[[i]]
100
300
500
700
200
400
600
800
1000
xa[[i]]

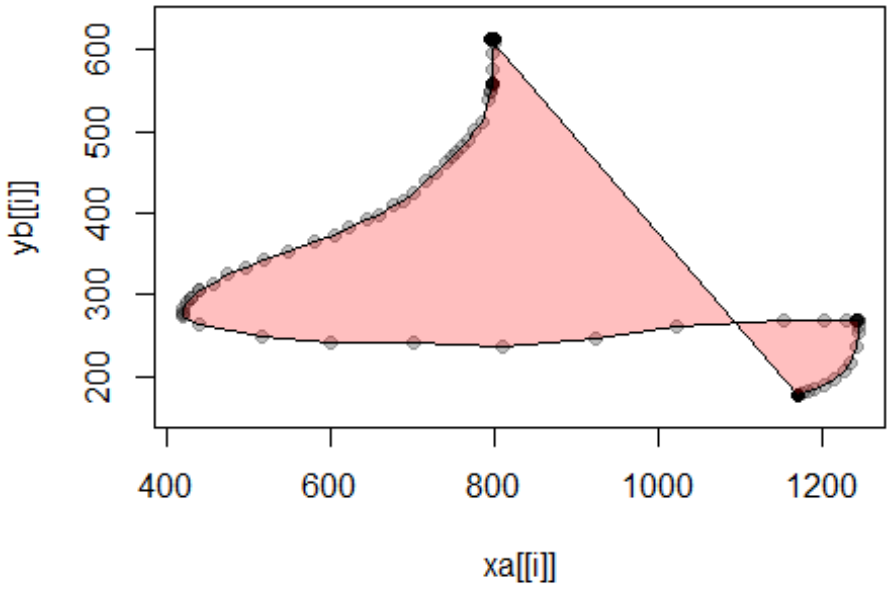

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

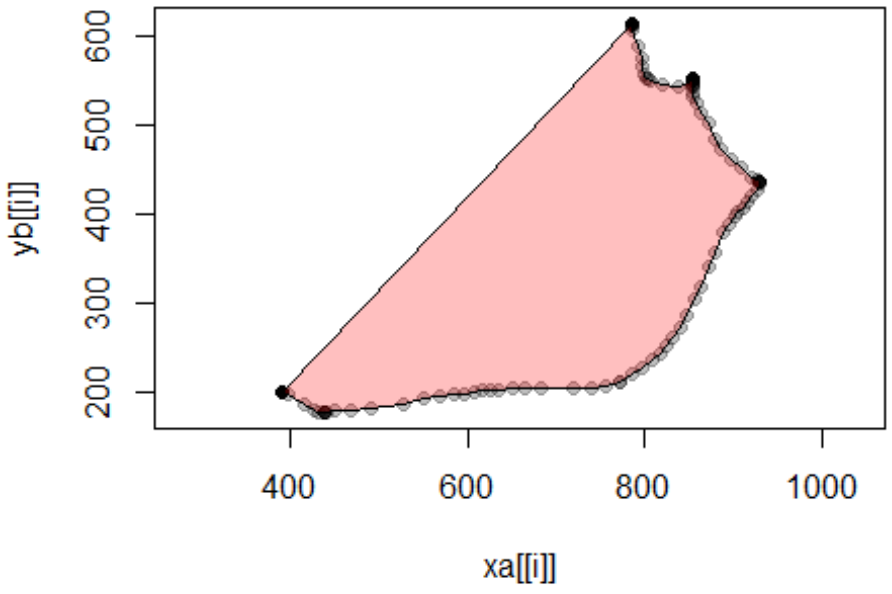

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

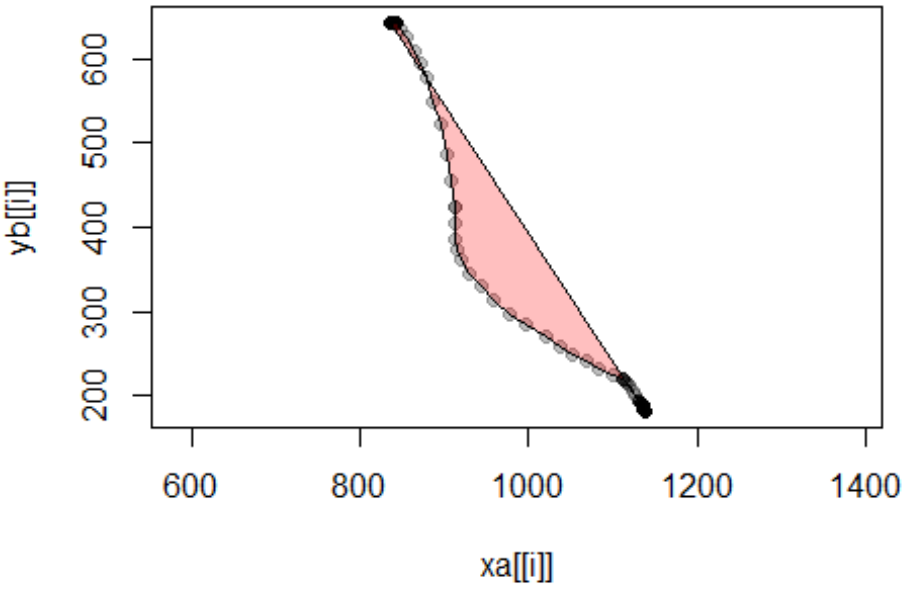

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

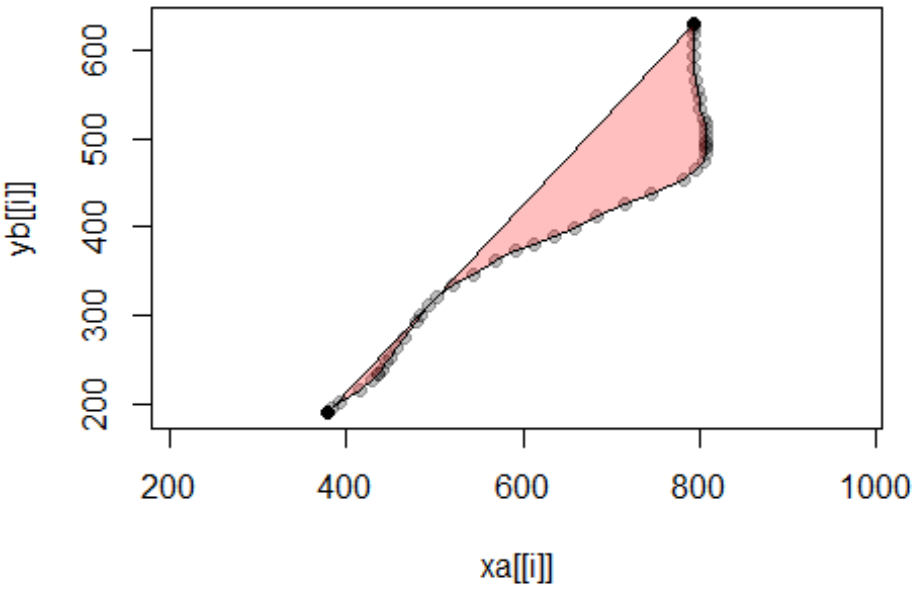

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

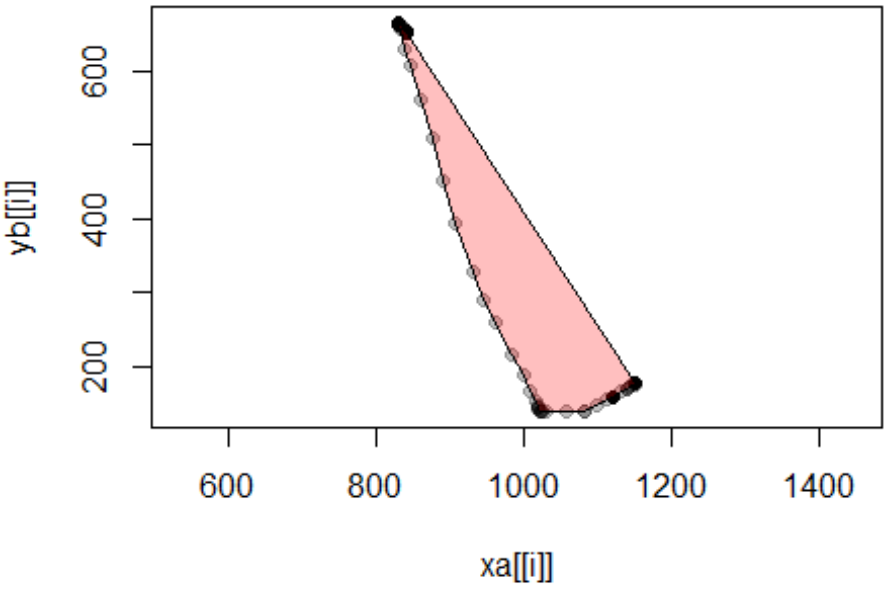

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

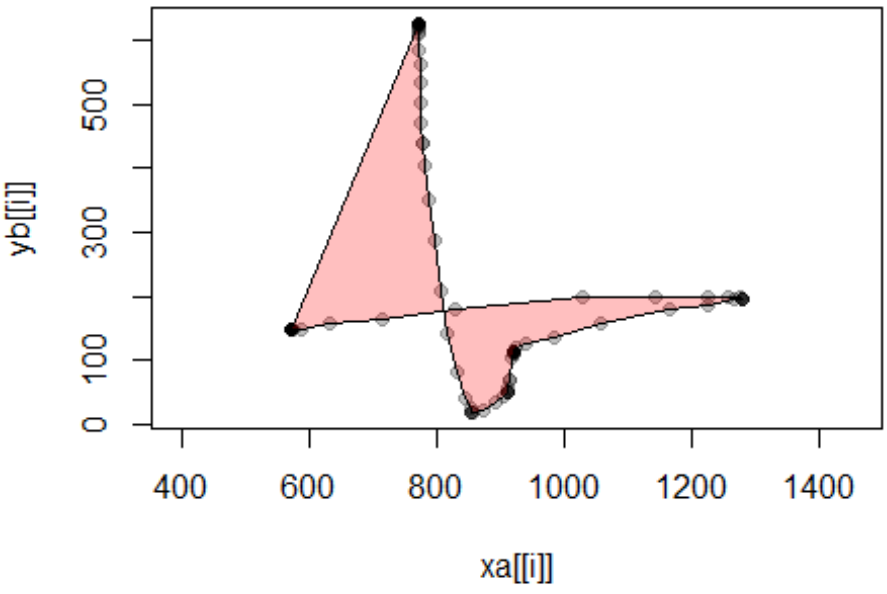

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

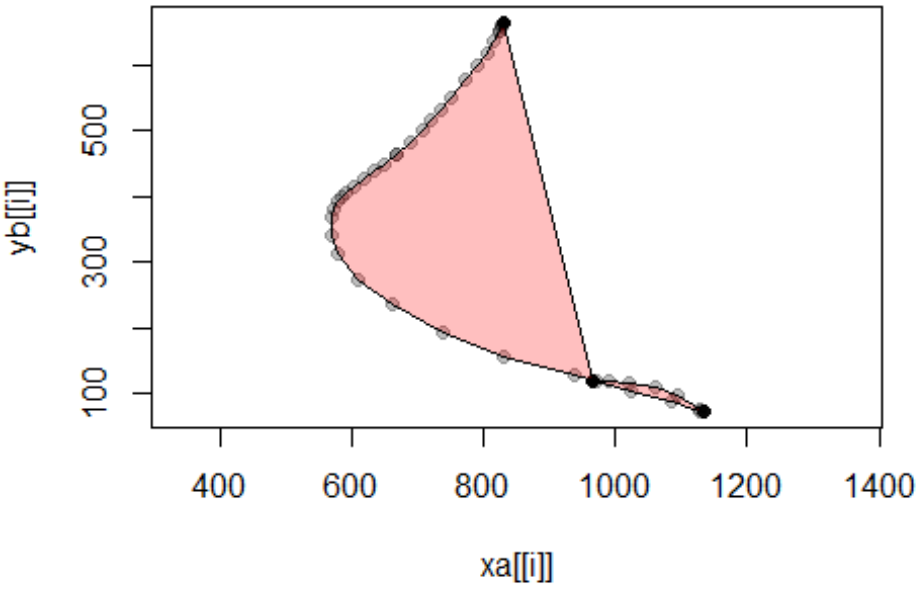

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

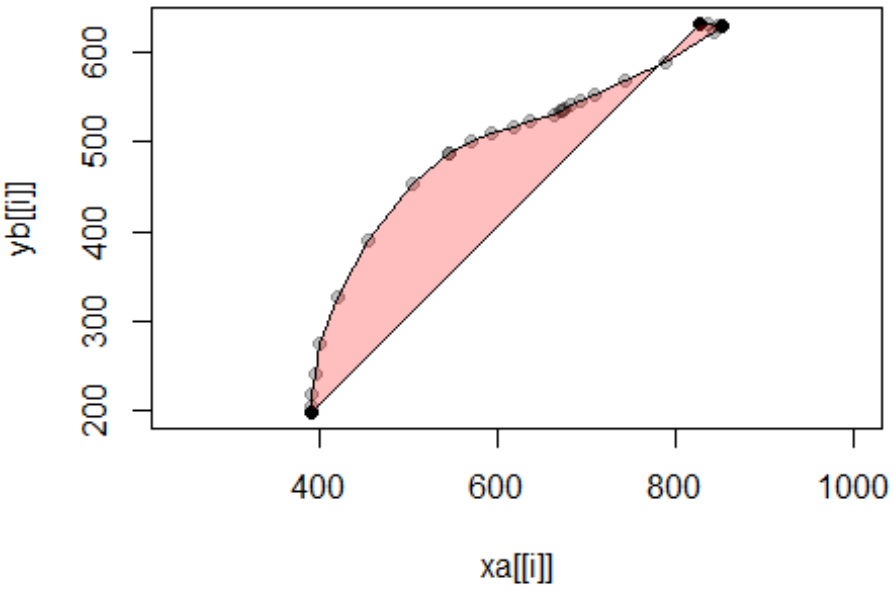

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

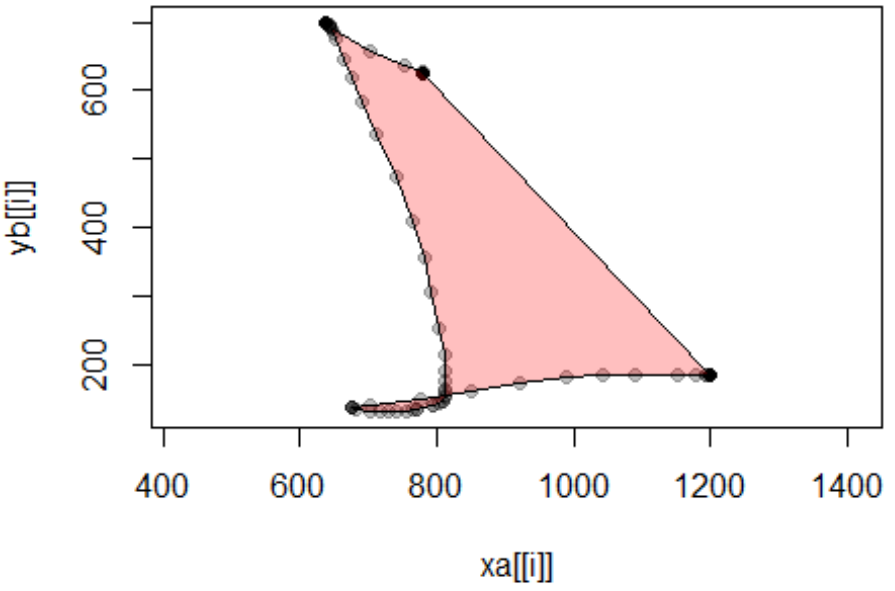

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

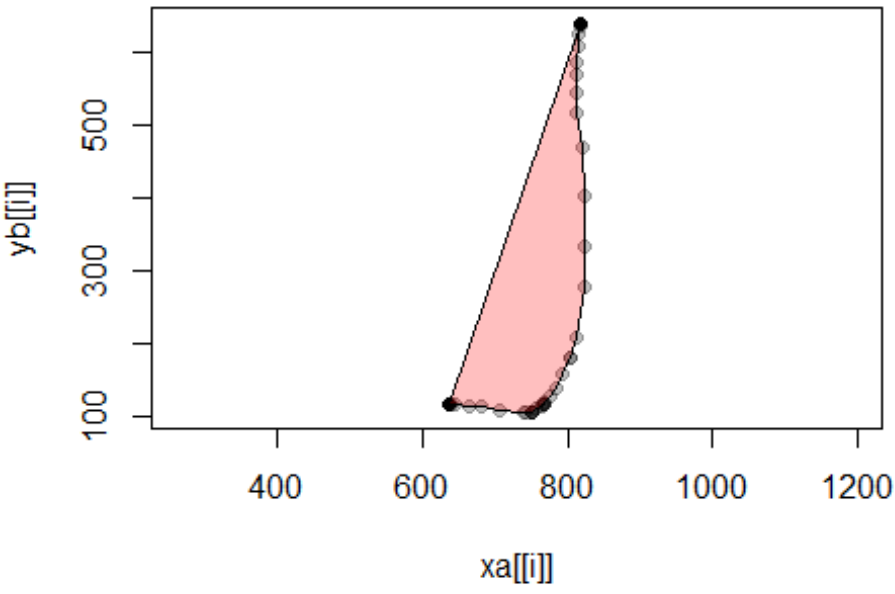

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

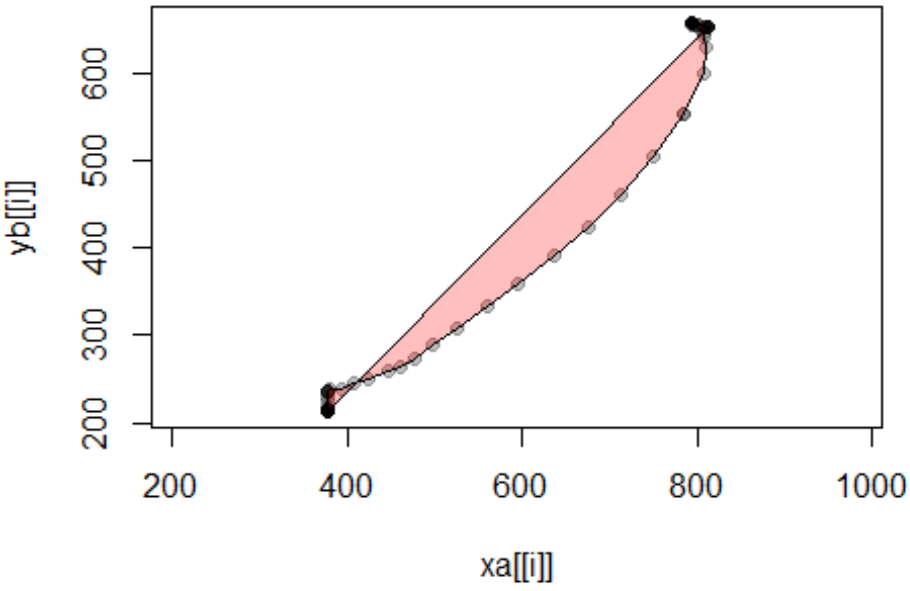

yb[[i]]
0
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

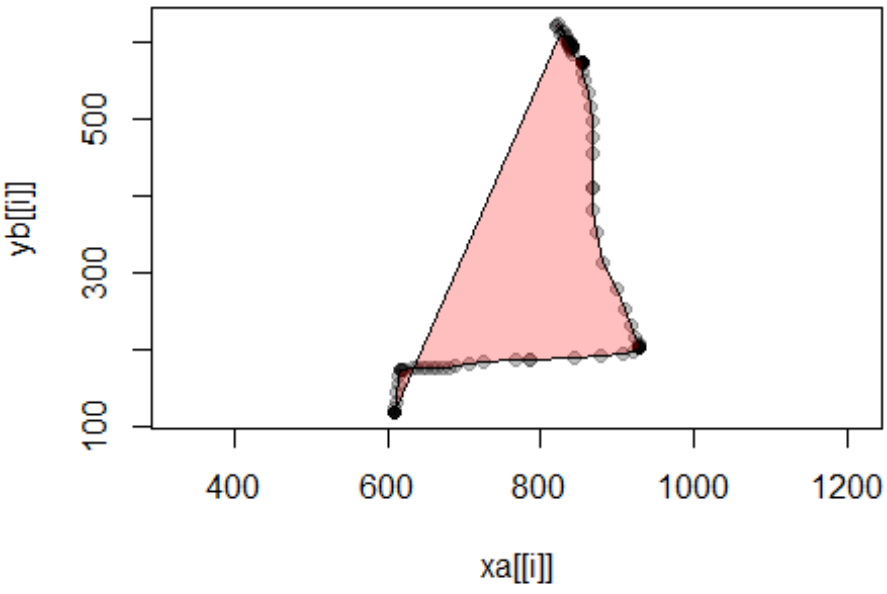

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

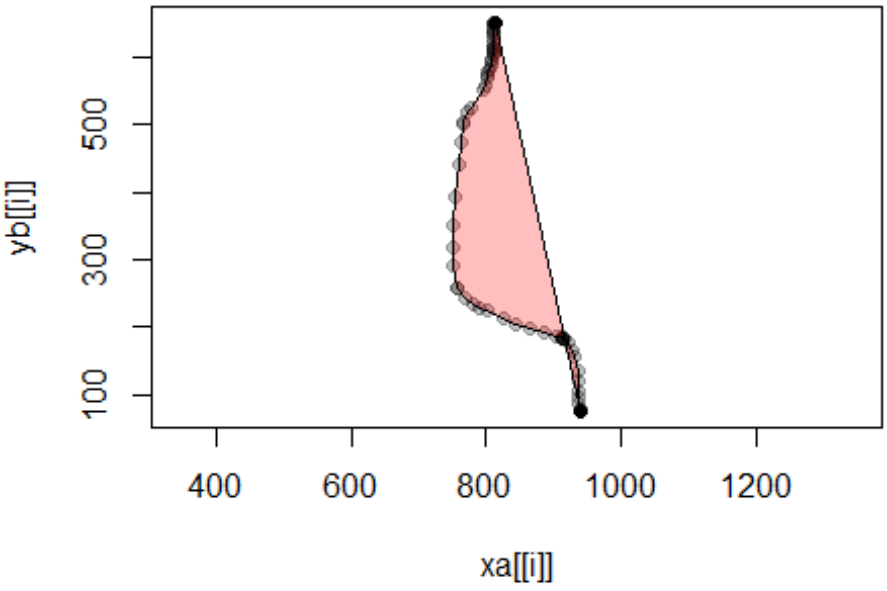

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

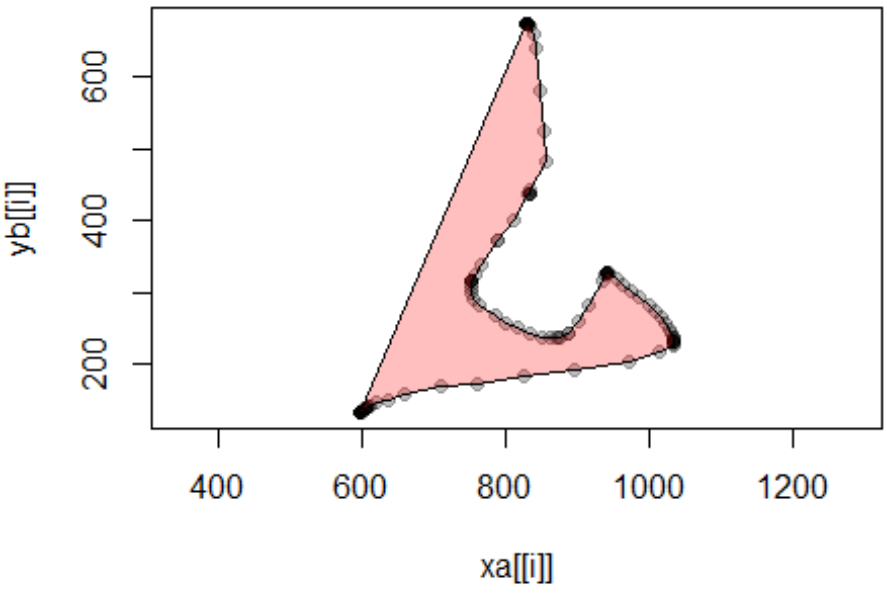

yb[[i]]
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

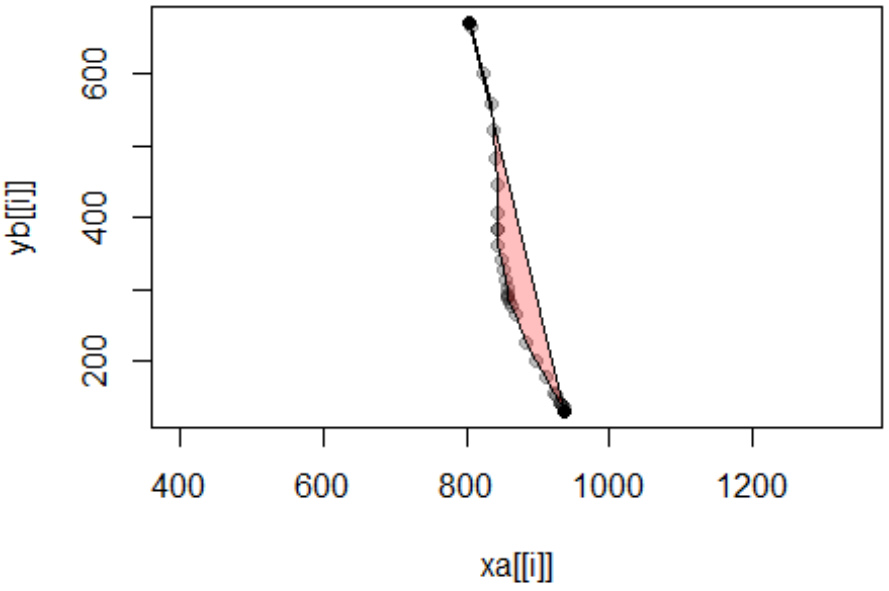

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

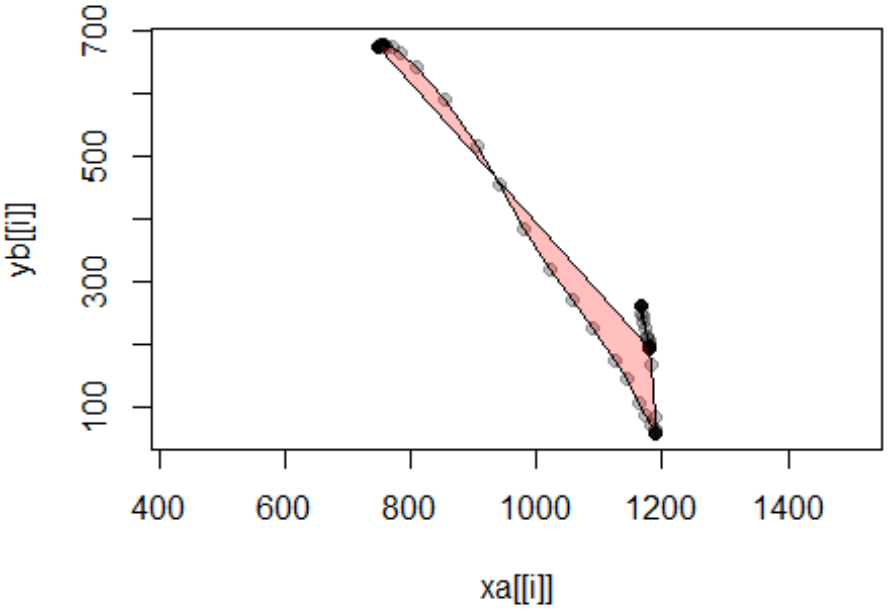

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

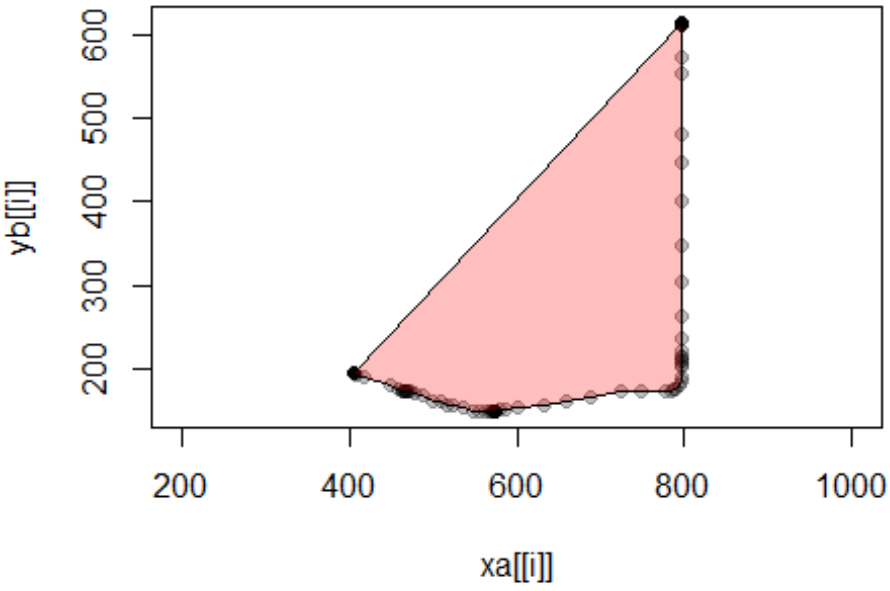

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

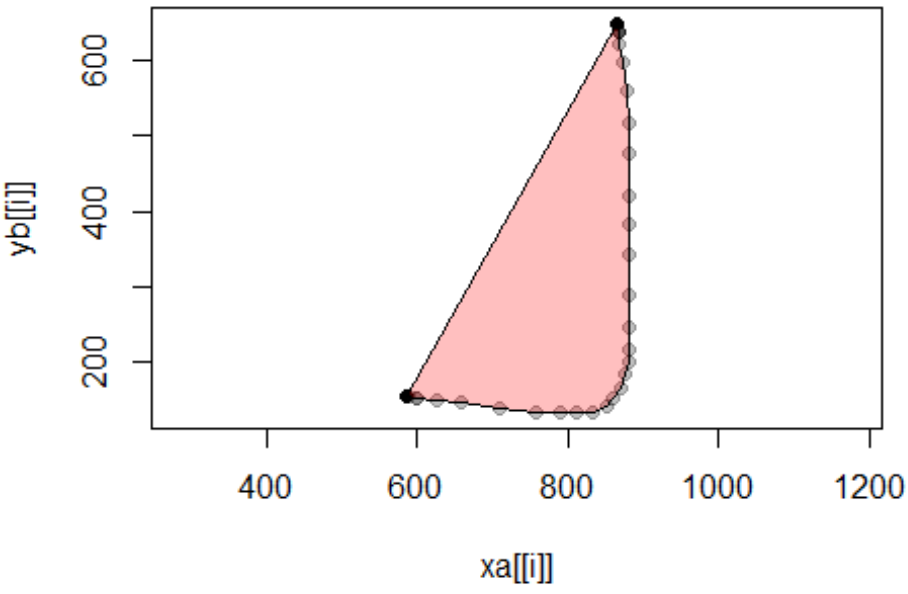

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

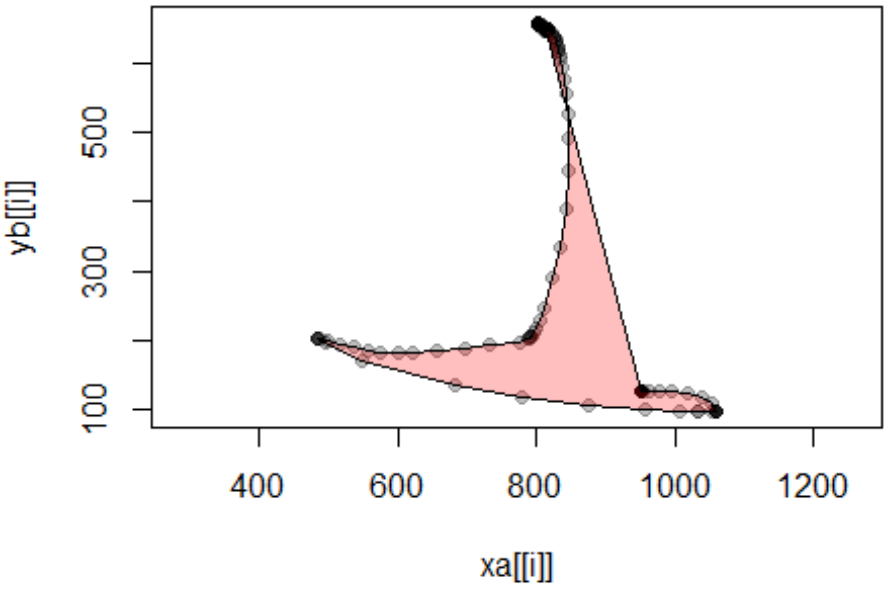

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

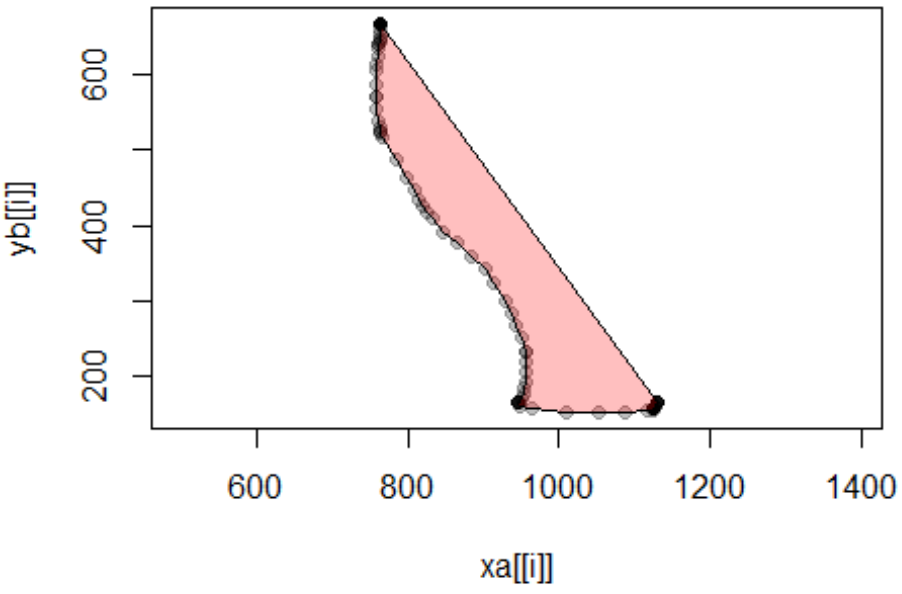

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

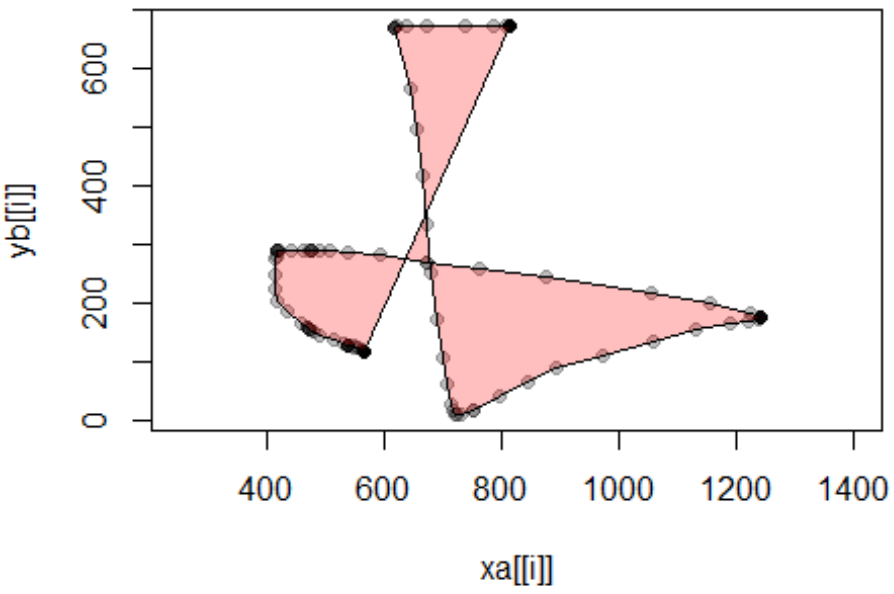

yb[[i]]
100
300
500
700
400
600
800
1000
1200
1400
xa[[i]]

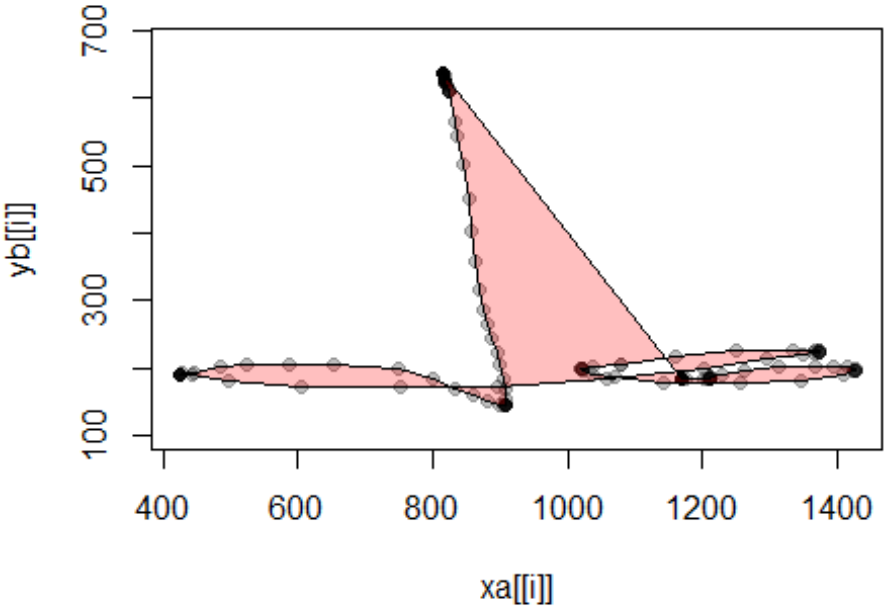

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

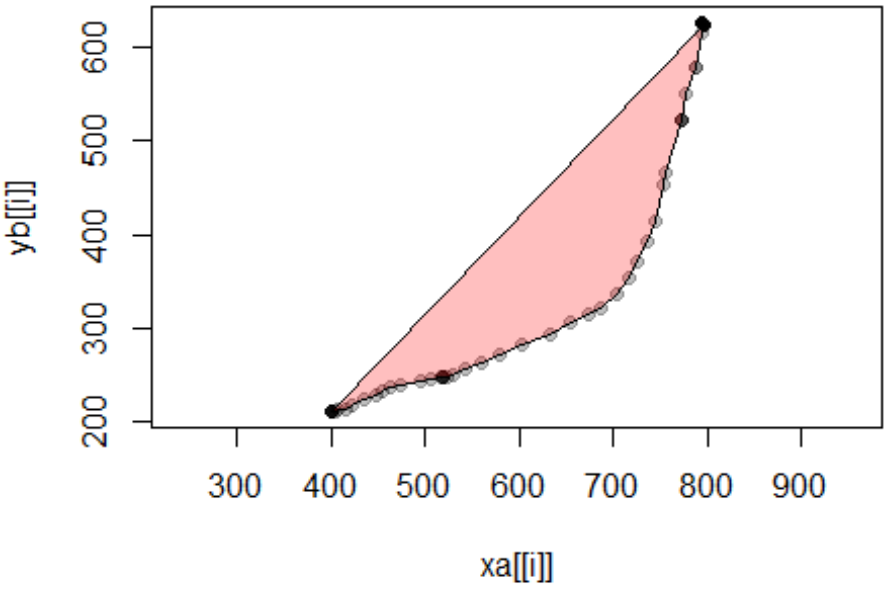

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

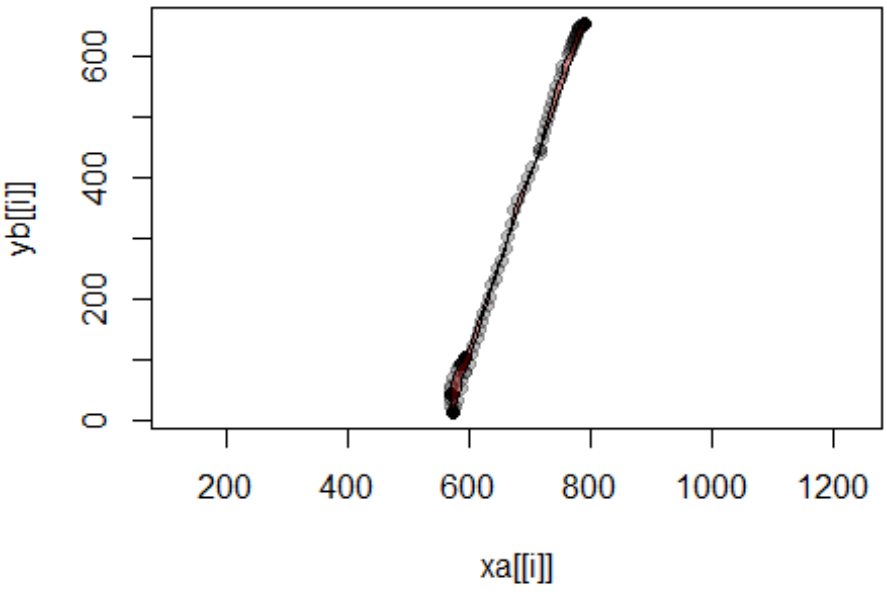

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

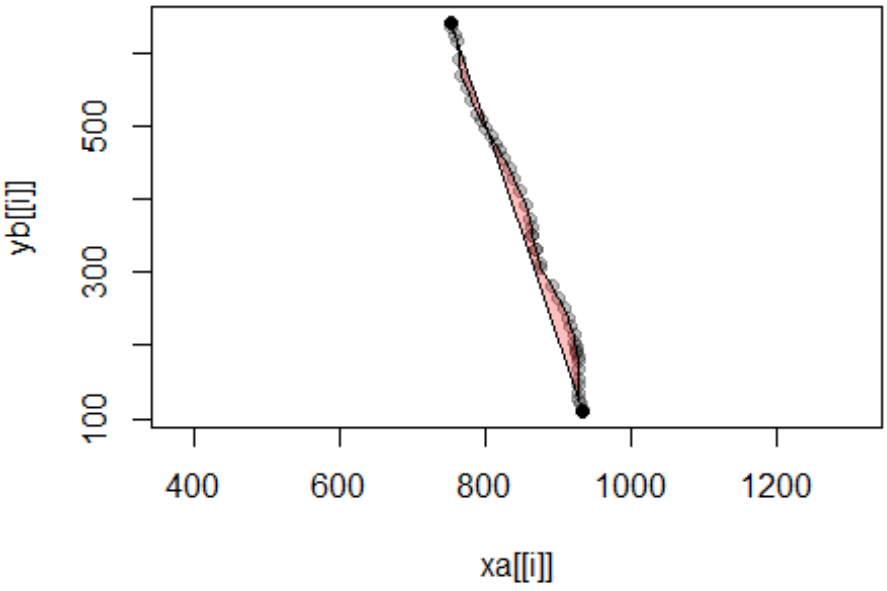

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

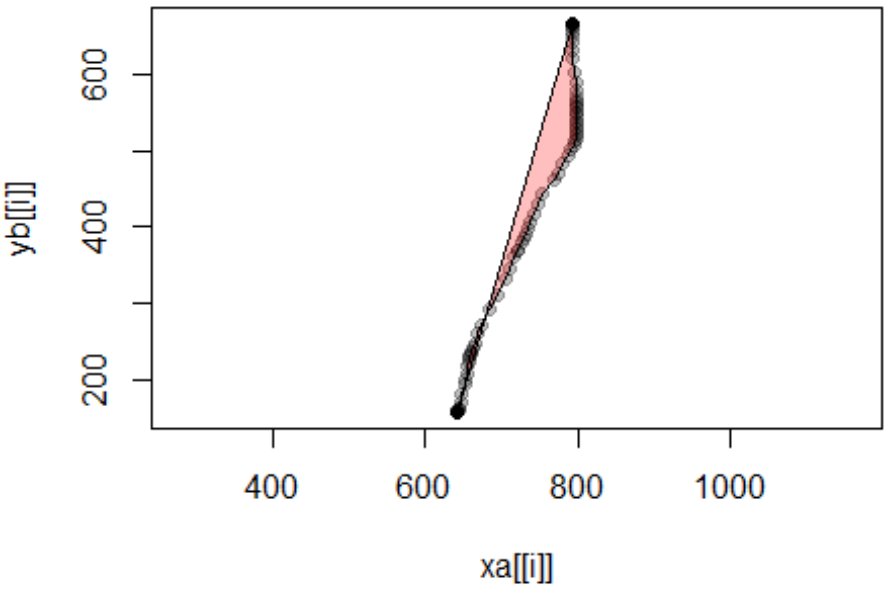

yb[[i]]
0
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

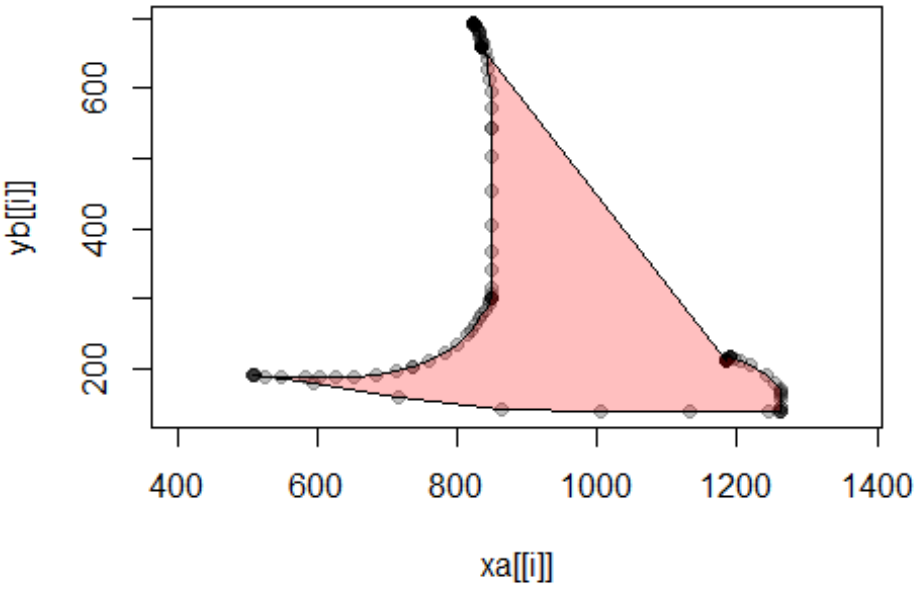

yb[[i]]
100
300
500
700
400
600
800
1000
1200
1400
xa[[i]]

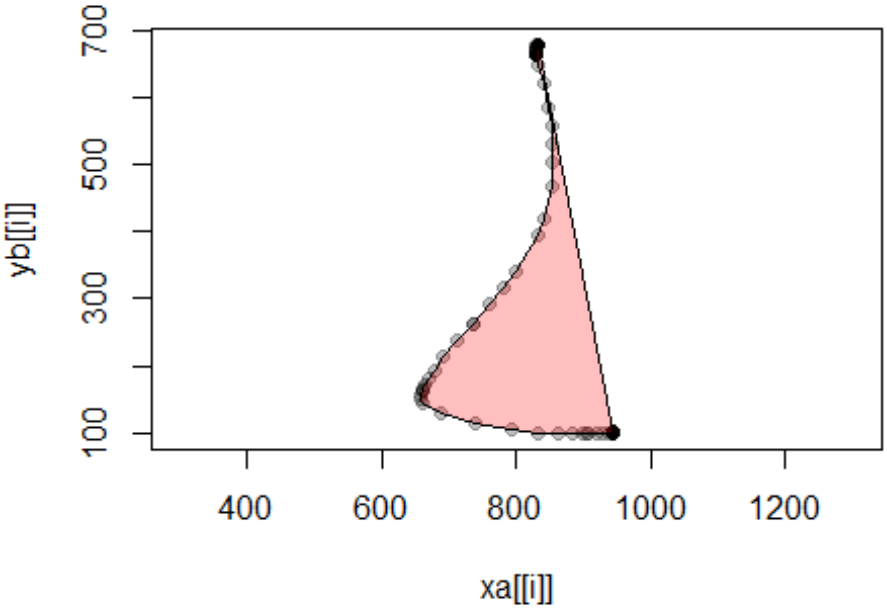

yb[[i]]
200
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

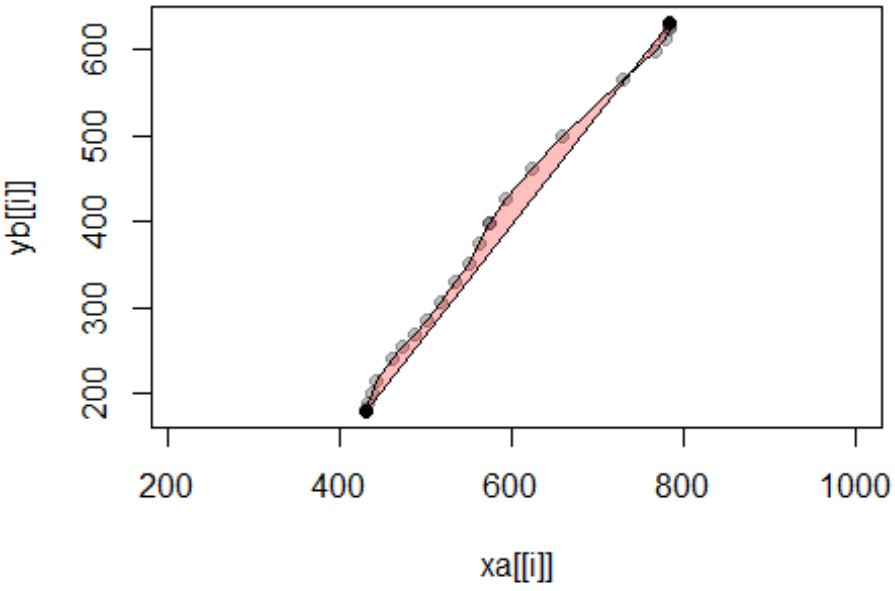

yb[[i]]
0
200
400
600
200
400
600
800
1000
1200
xa[[i]]

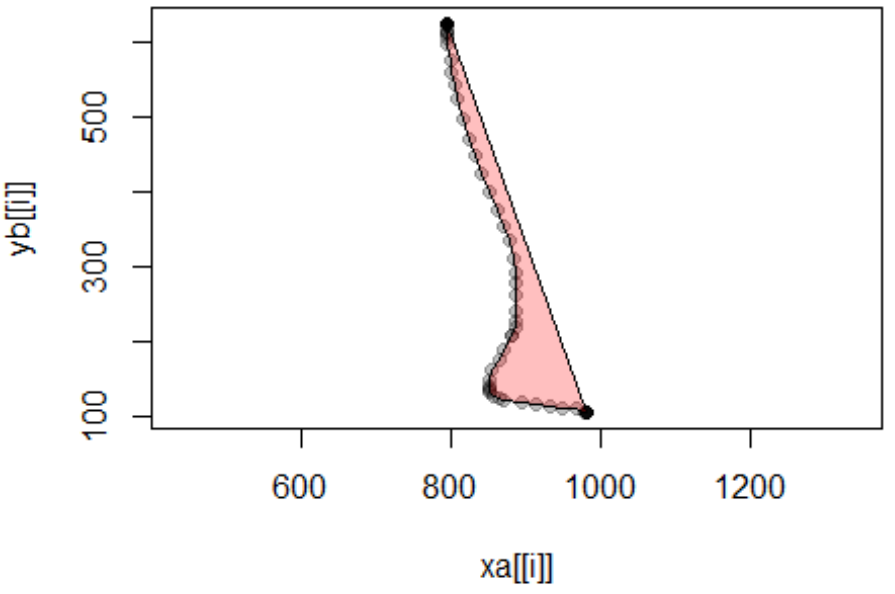

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

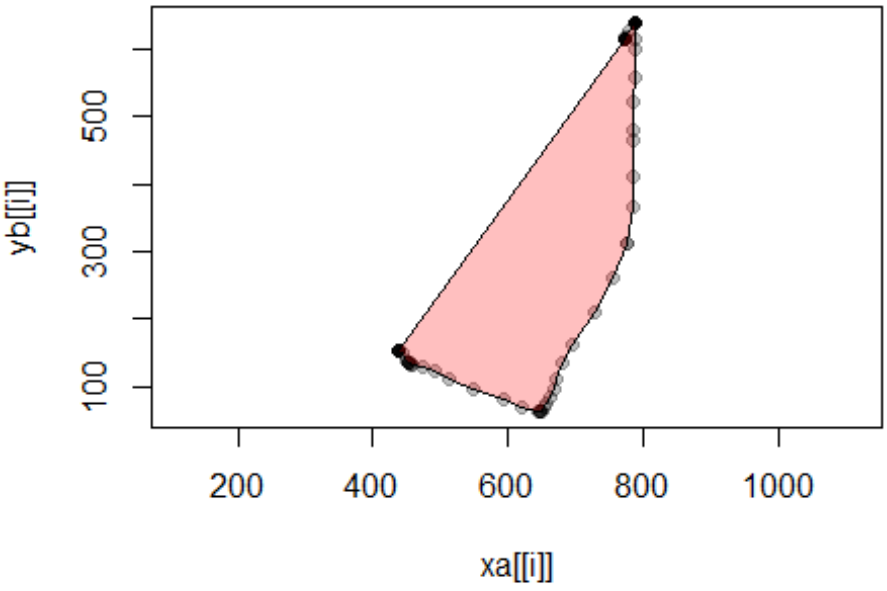

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

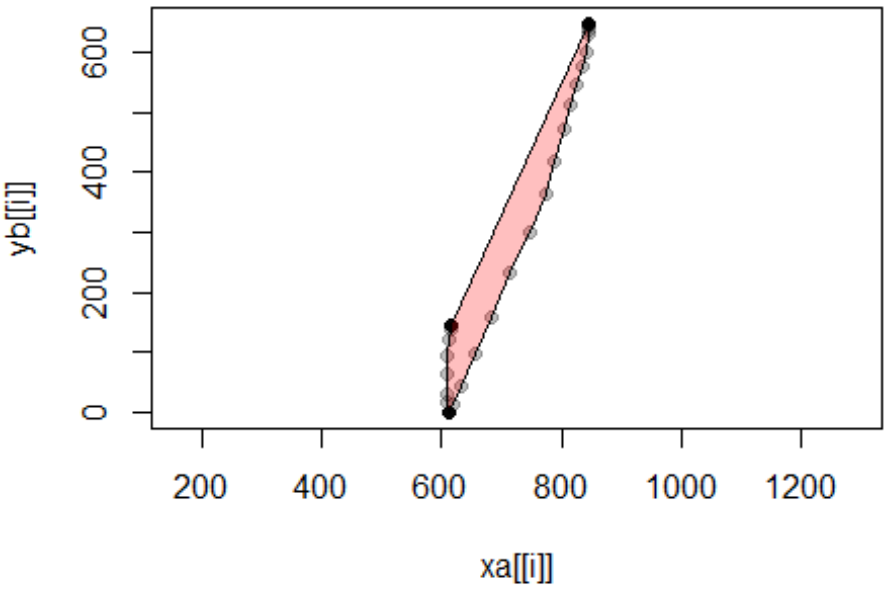

yb[[i]]
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

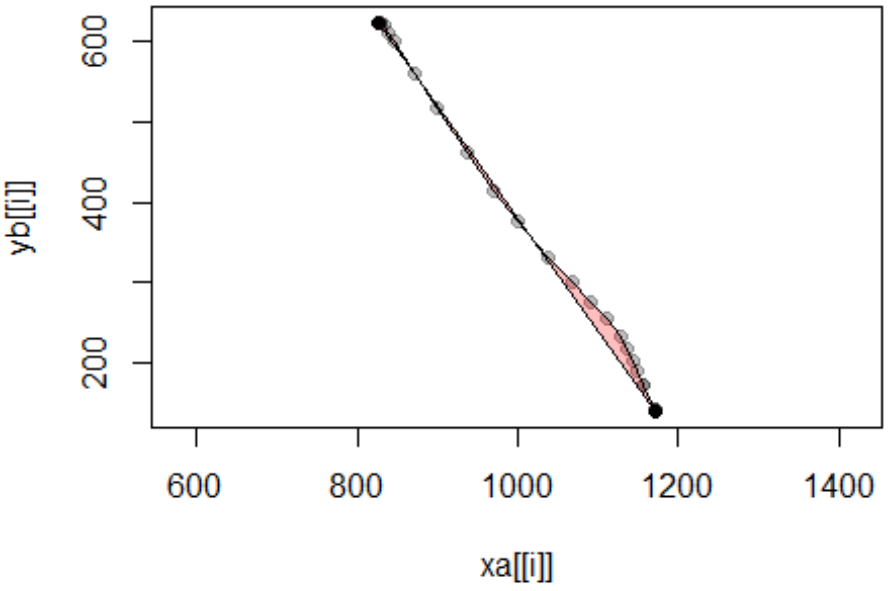

yb[[i]]
100
300
500
700
400
600
800
1000
1200
xa[[i]]

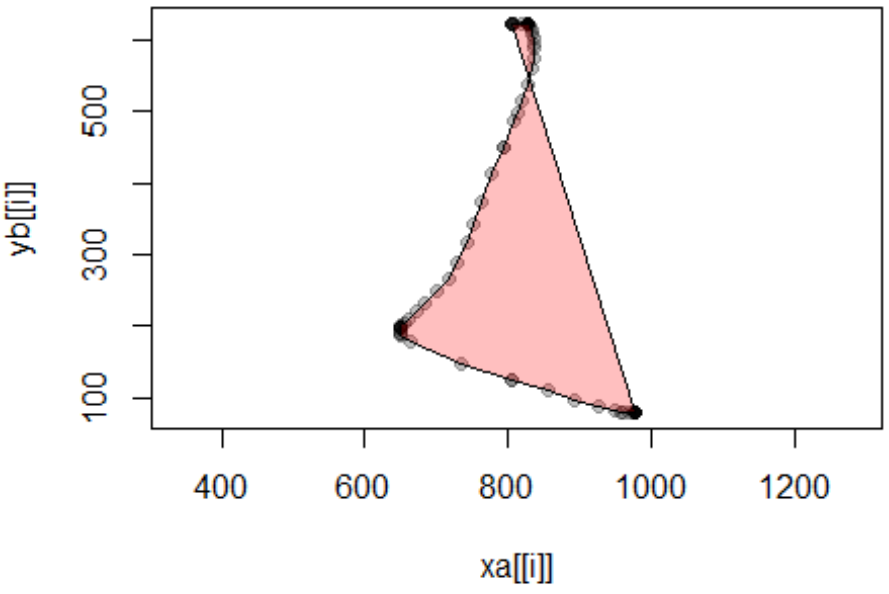

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

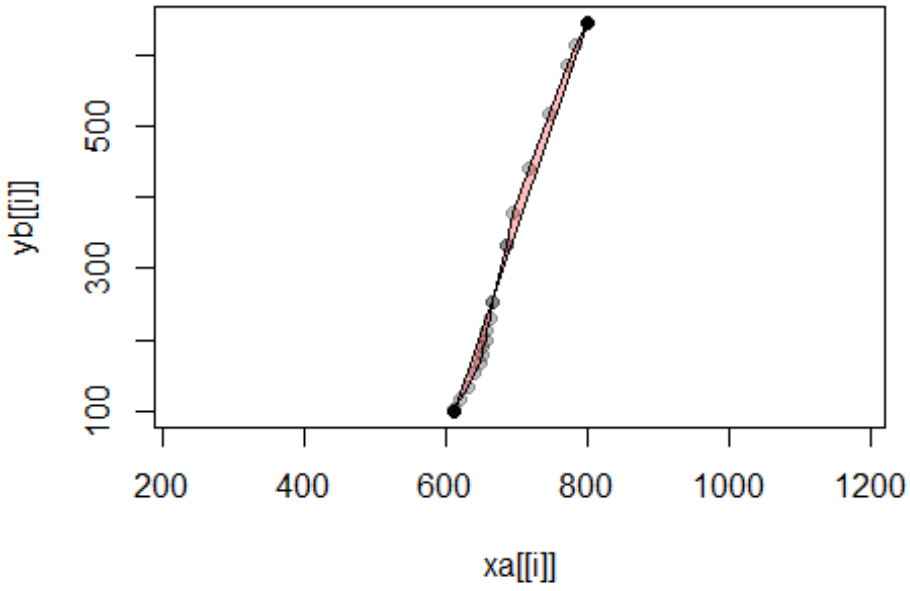

yb[[i]]
100
300
500
600
800
1000
1200
xa[[i]]

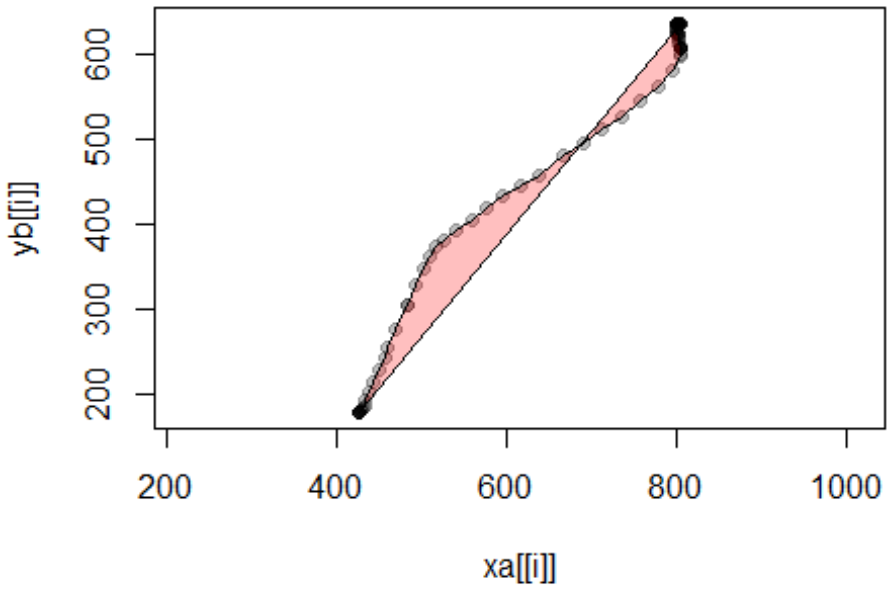

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

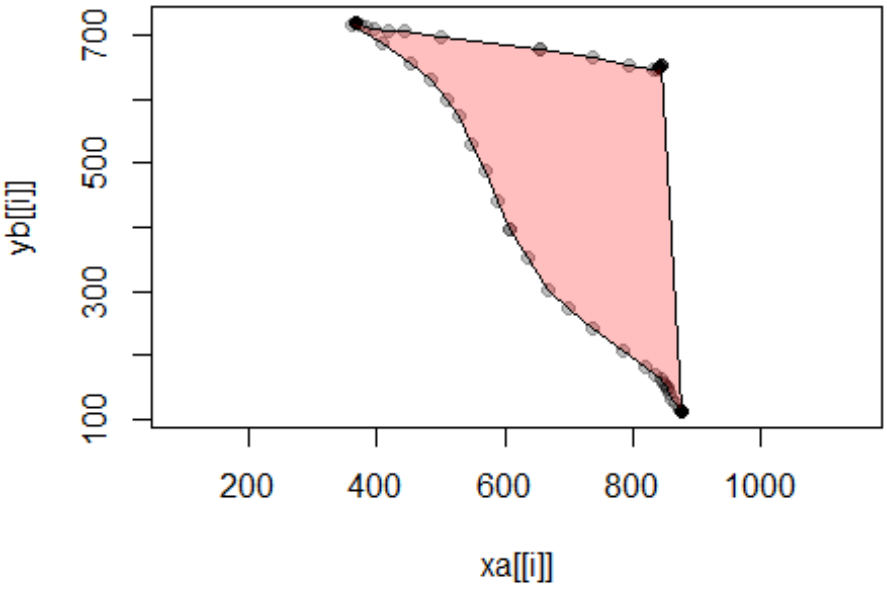

yb[[i]]
0
200
400
600
200
400
600
800
1000
1200
xa[[i]]

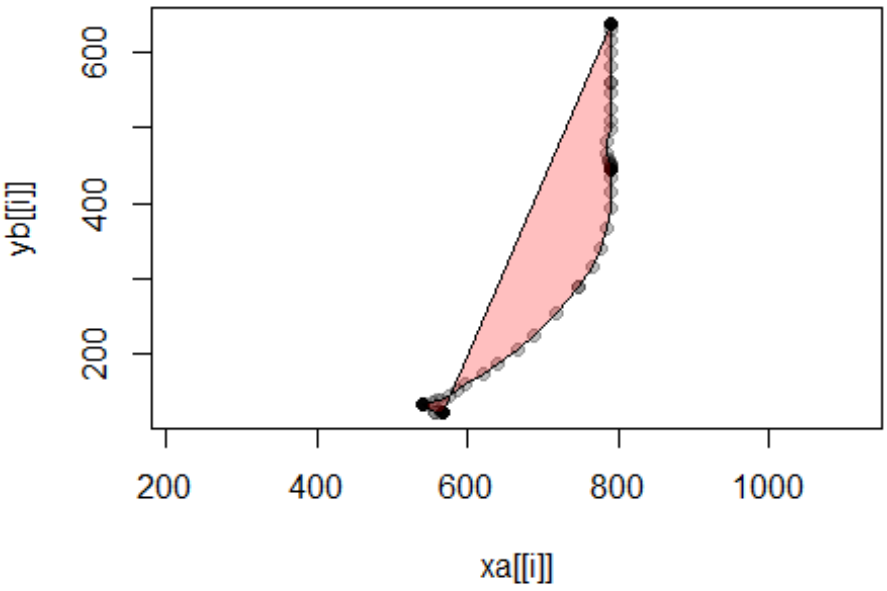

yb[[i]]
600
400
200
600
800
1000
1200
1400
xa[[i]]

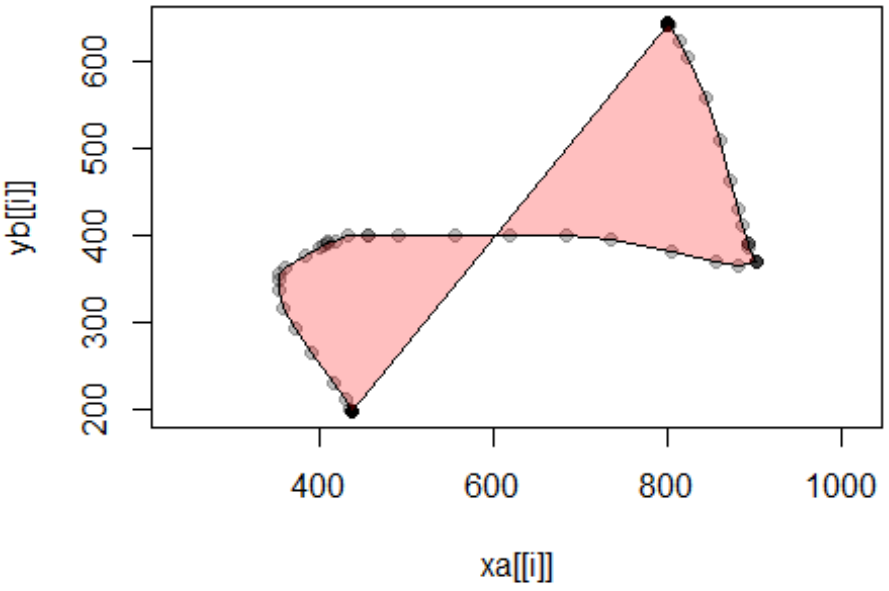

yb[[i]]
500
300
100
400
600
800
1000
1200
xa[[i]]

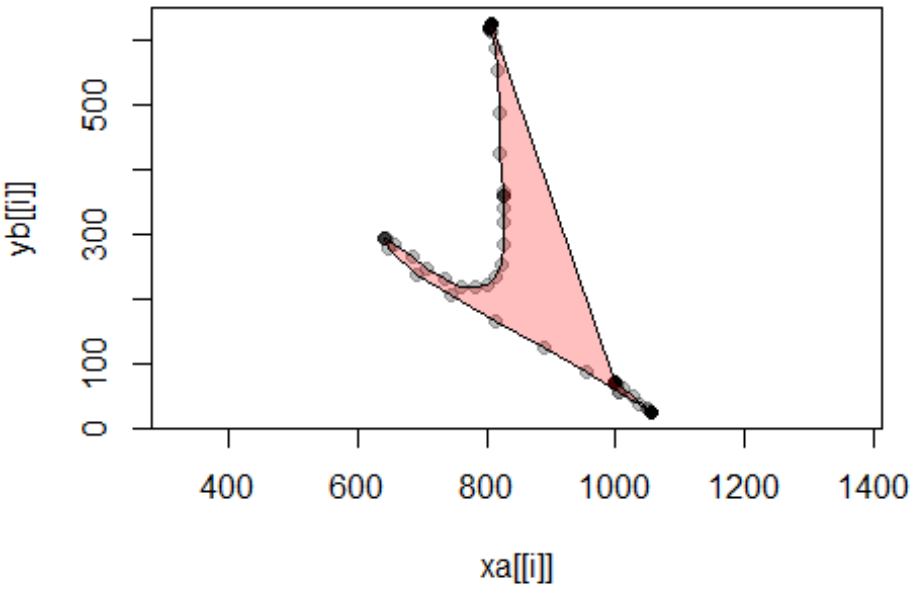

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

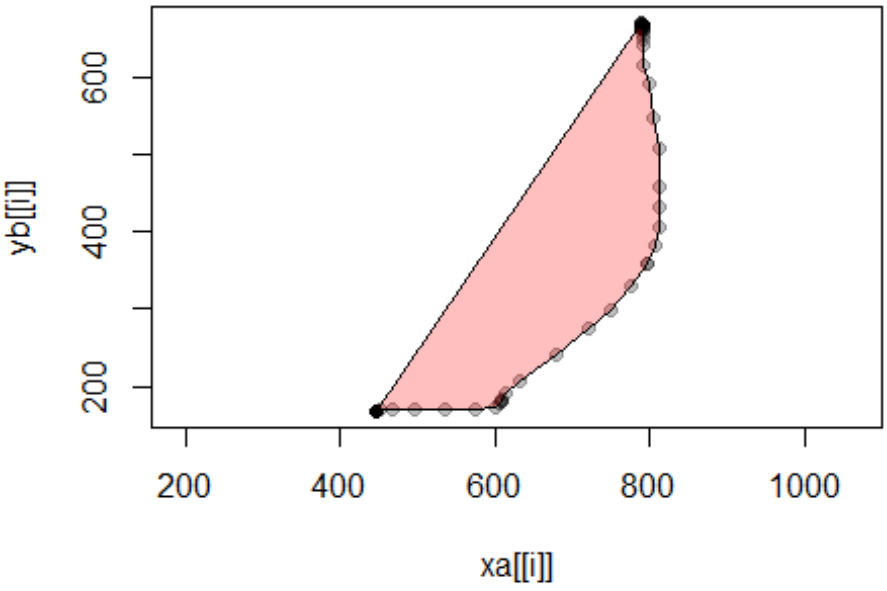

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

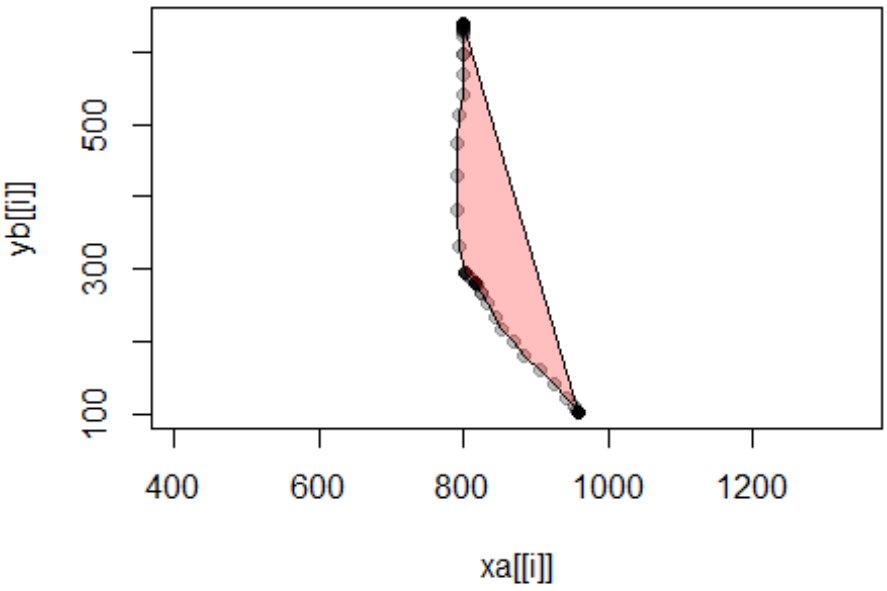

yb[[i]]
100
300
500
700
200
400
600
800
1000
xa[[i]]

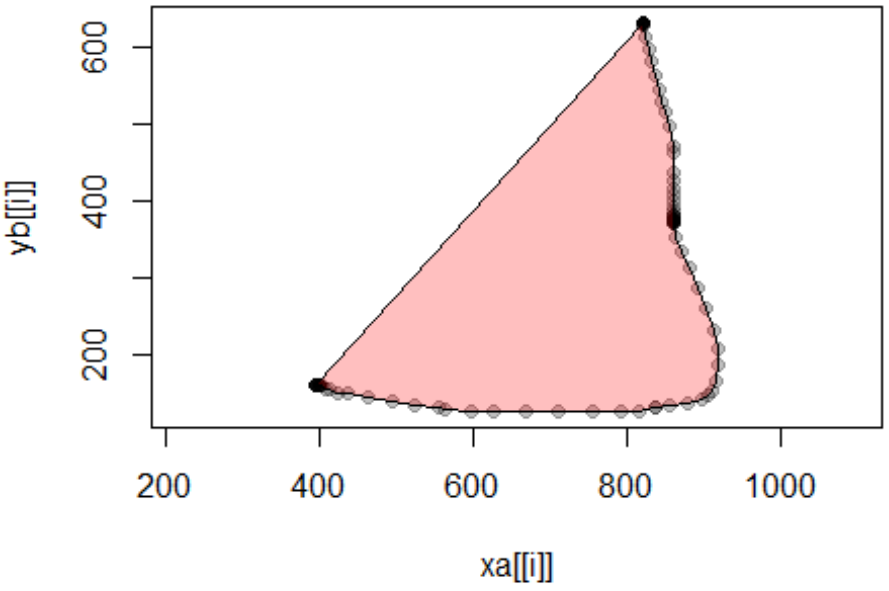

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

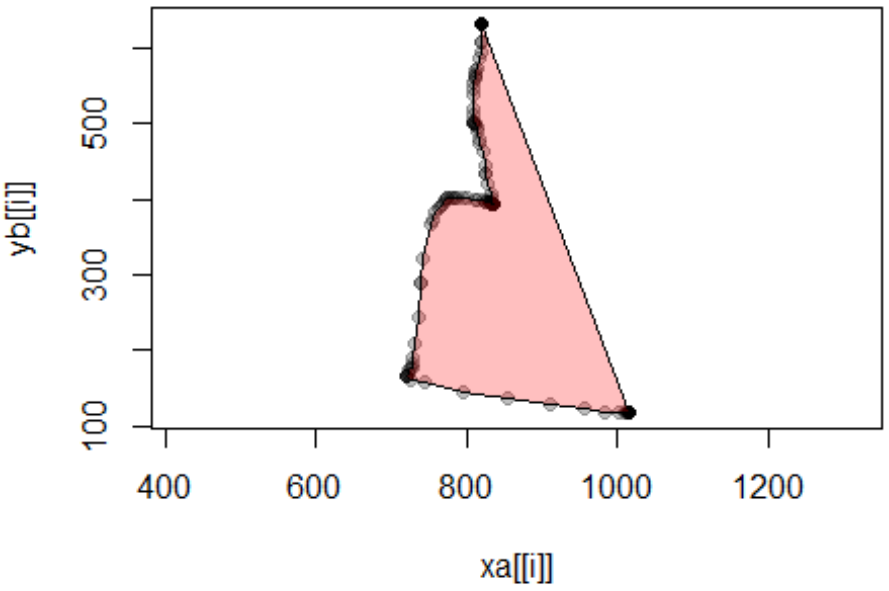

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

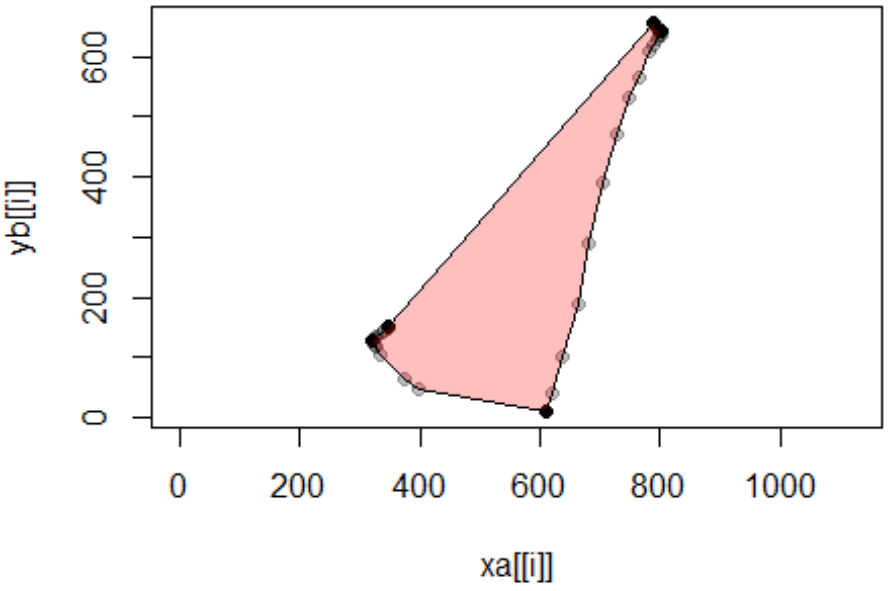

yb[[i]]
0
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

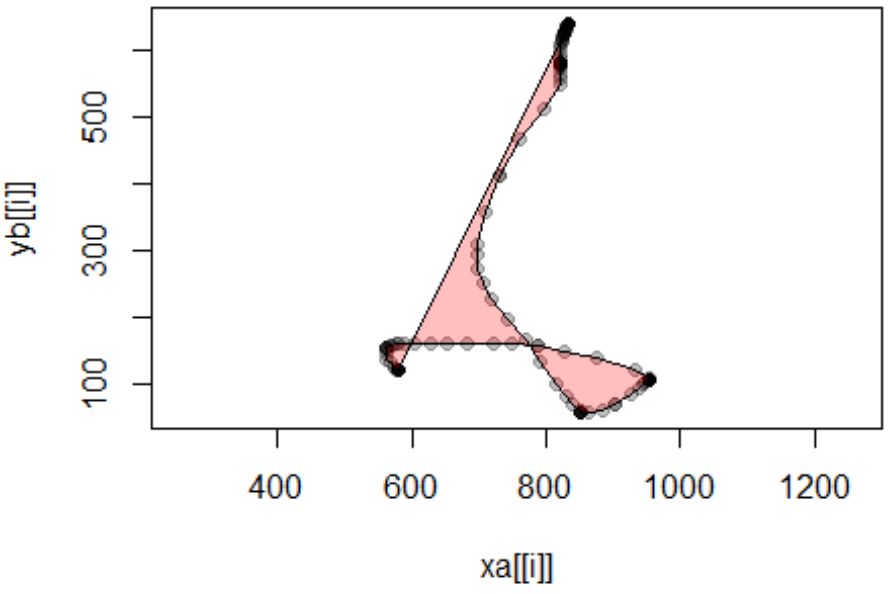

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

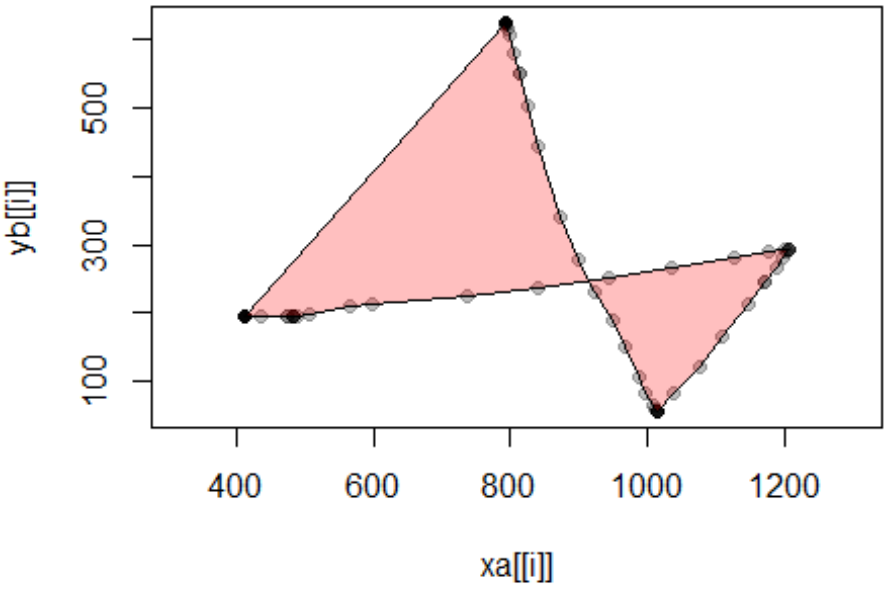

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

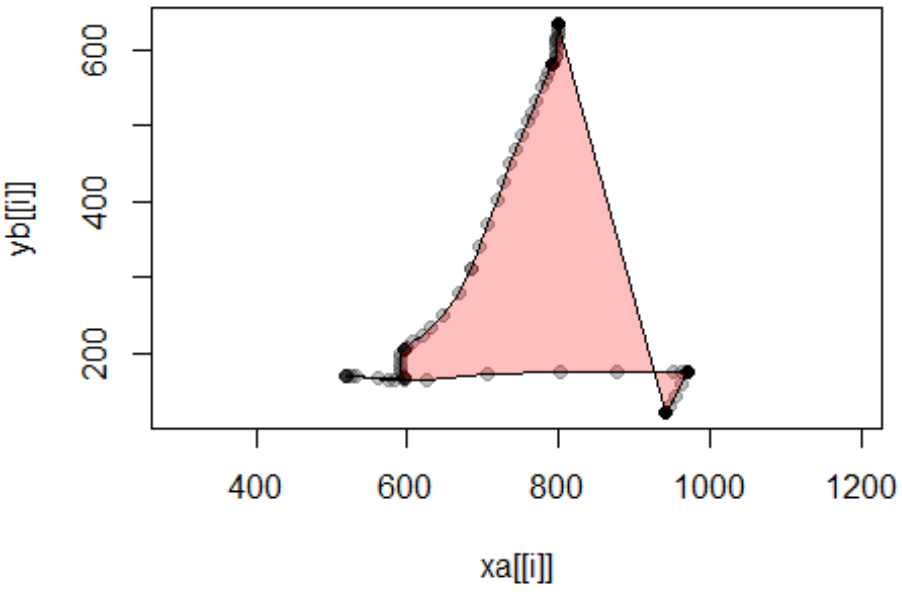

yb[[i]]
600
400
200
200
400
600
800
1000
xa[[i]]

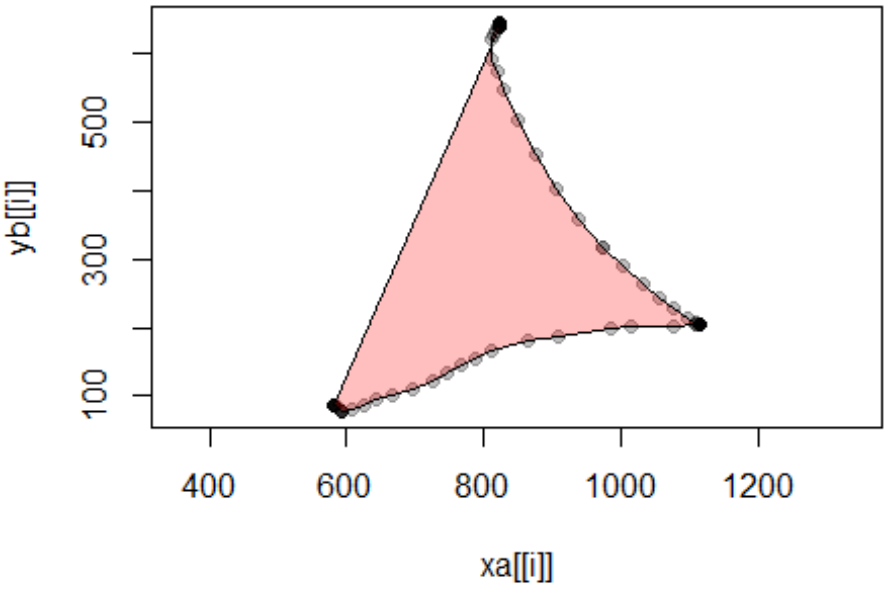

yb[[i]]
500
300
100
400
600
800
1000
1200
xa[[i]]

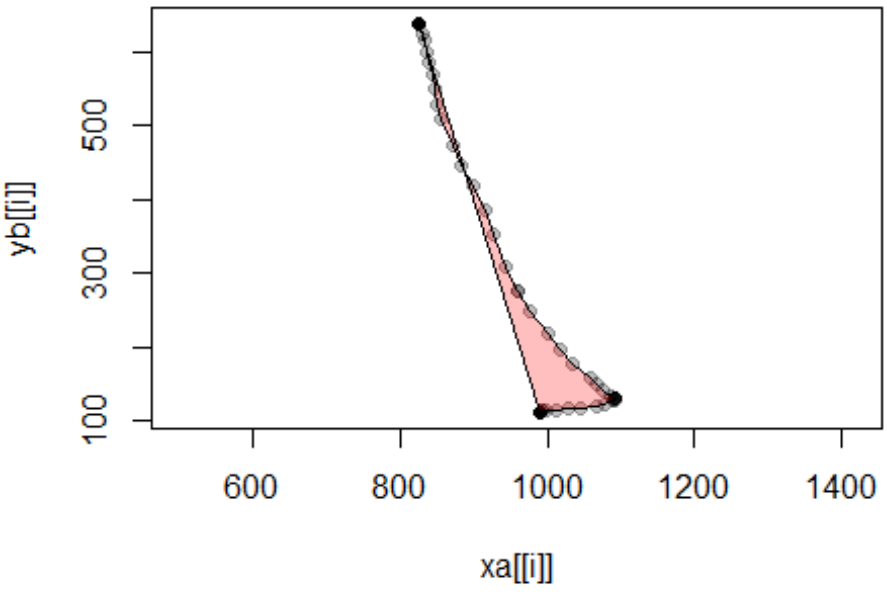

yb[[i]]
0
200
400
600
0
200
400
600
800
1000
xa[[i]]

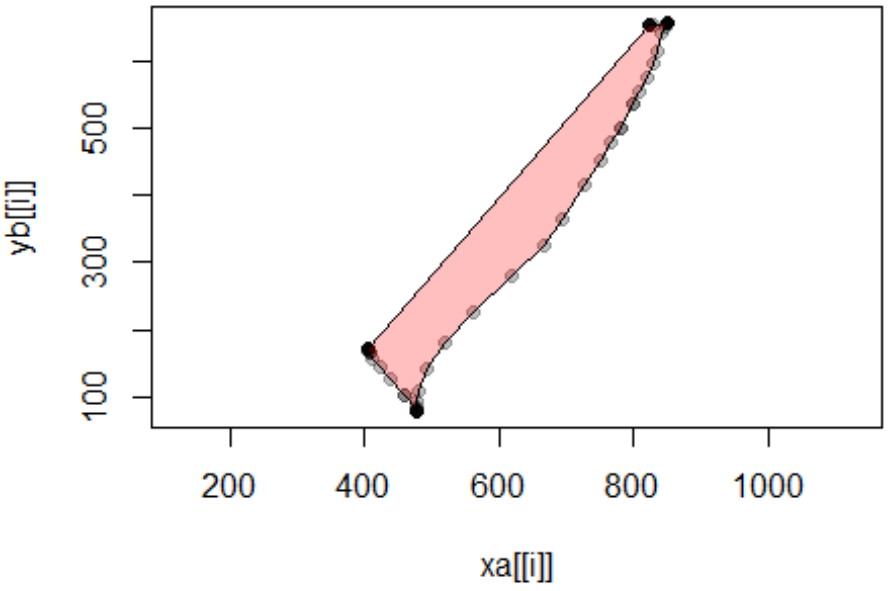

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

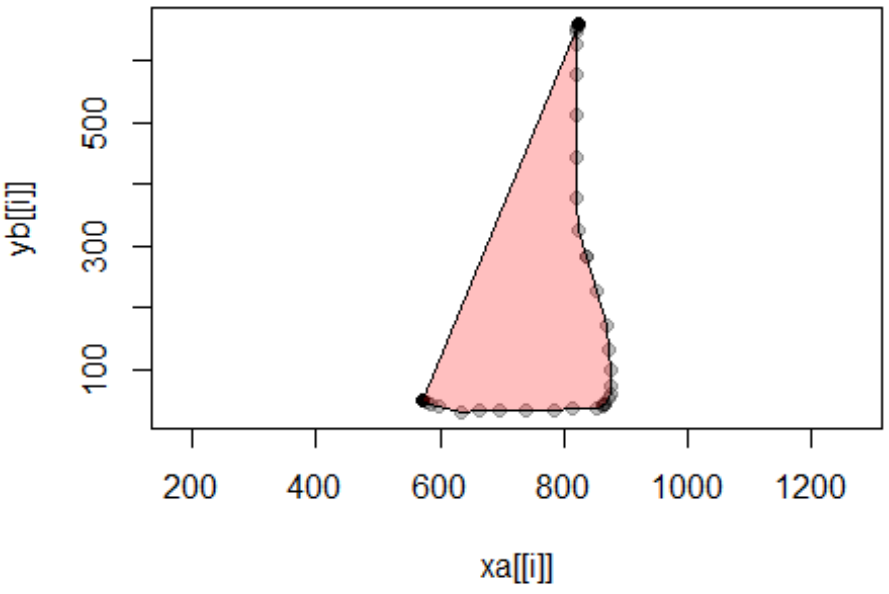

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

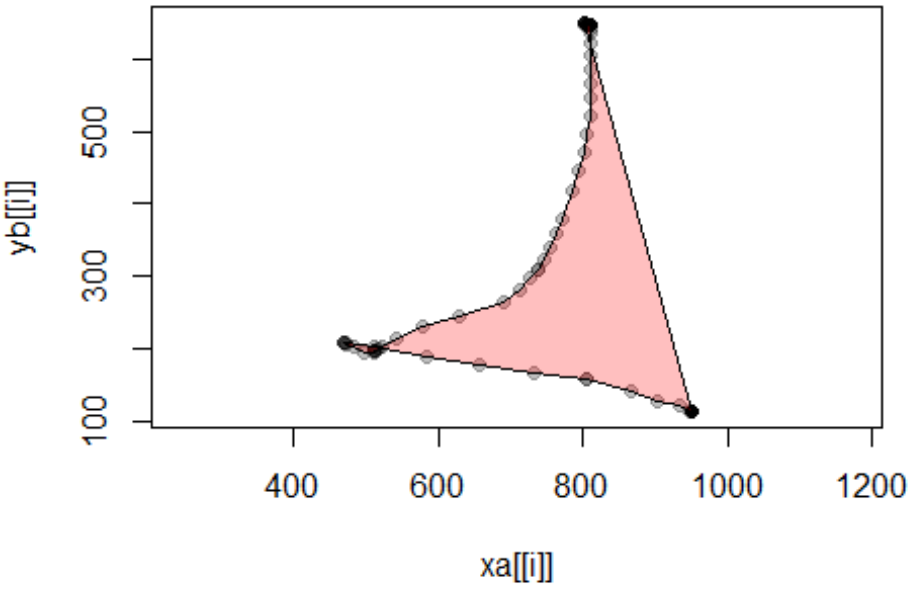

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

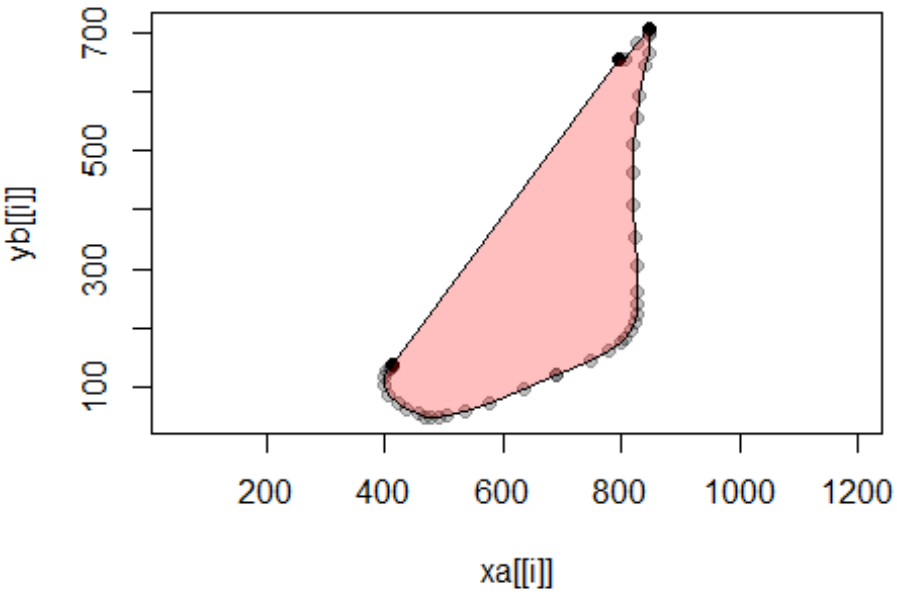

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

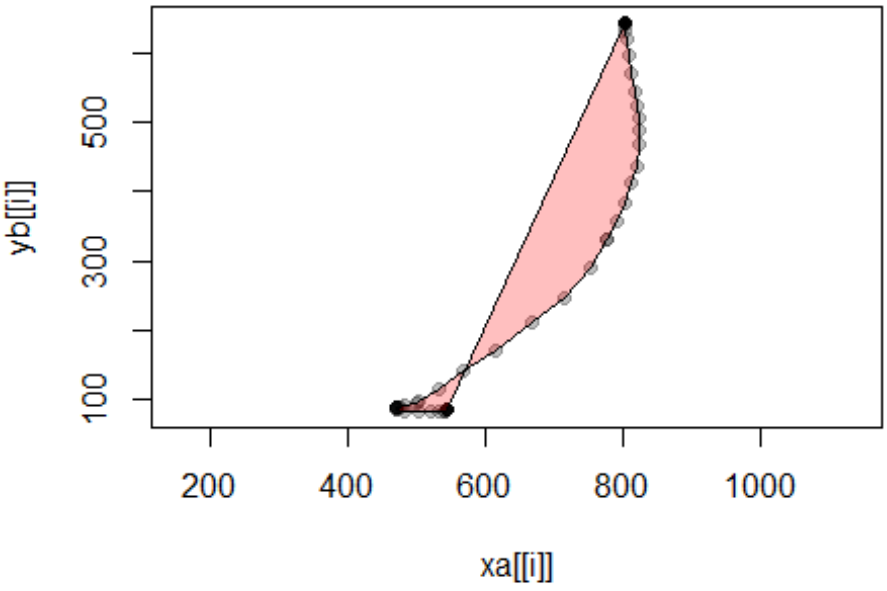

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

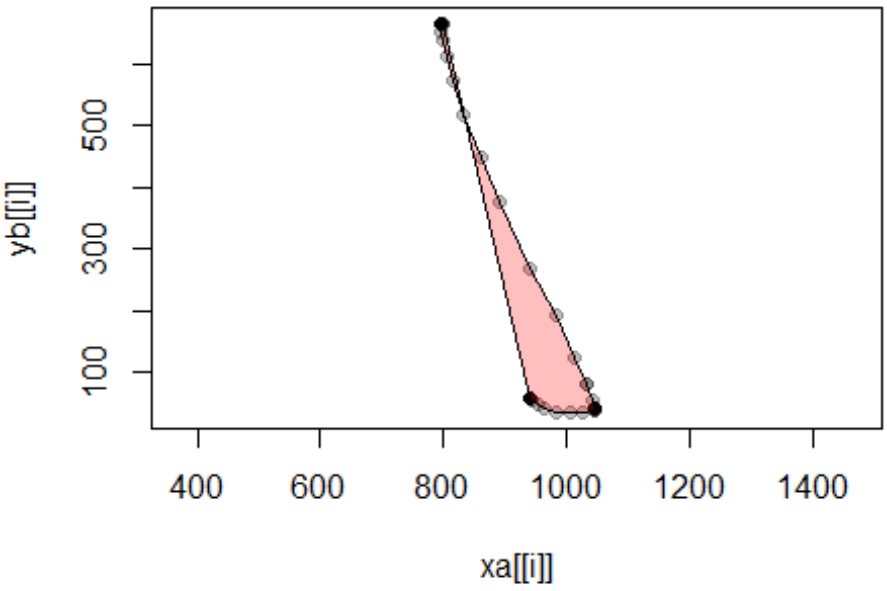

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

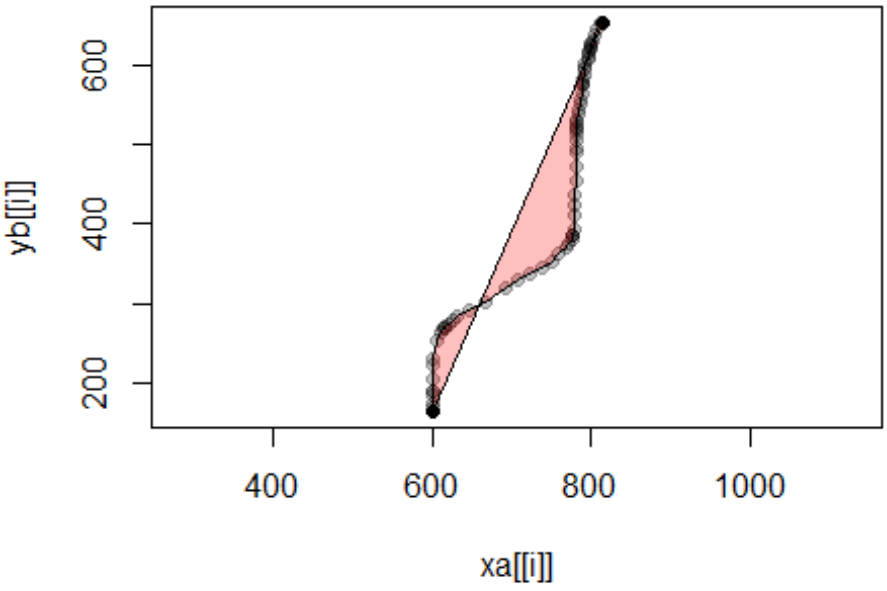

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

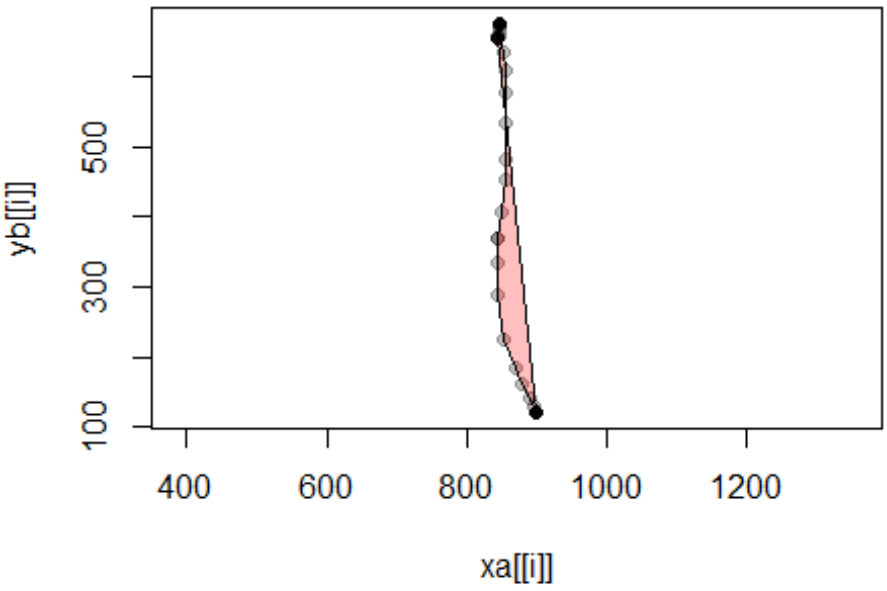

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

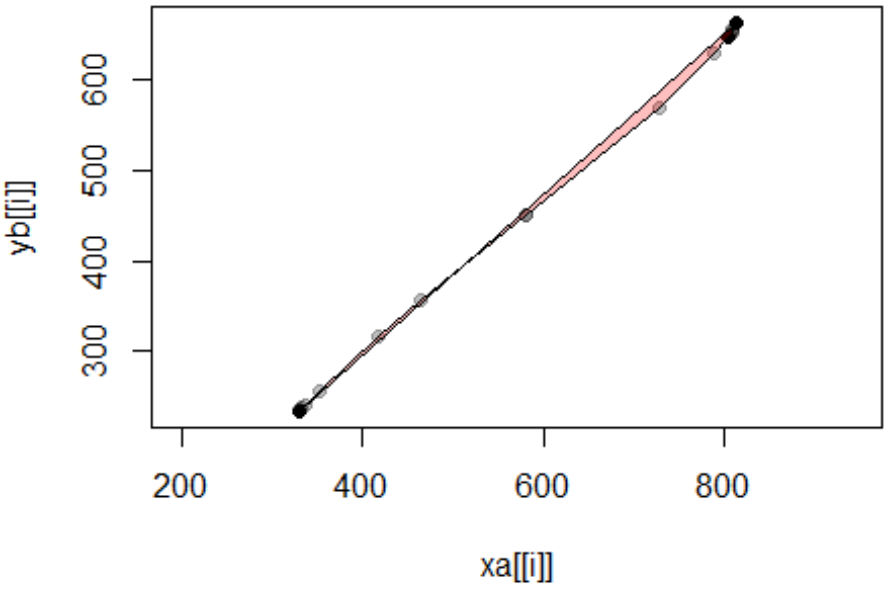

yb[[i]]
100
300
500
700
200
400
600
800
1000
1200
xa[[i]]

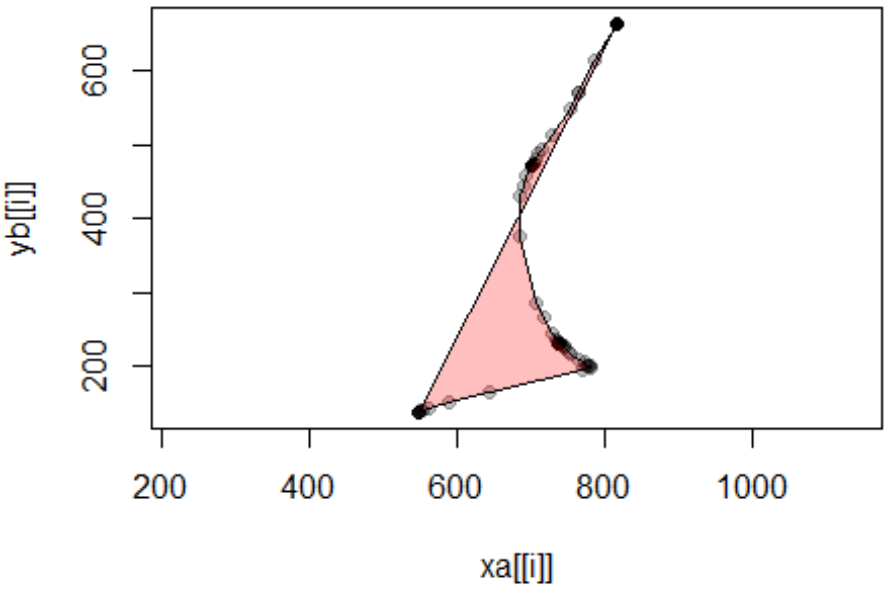

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

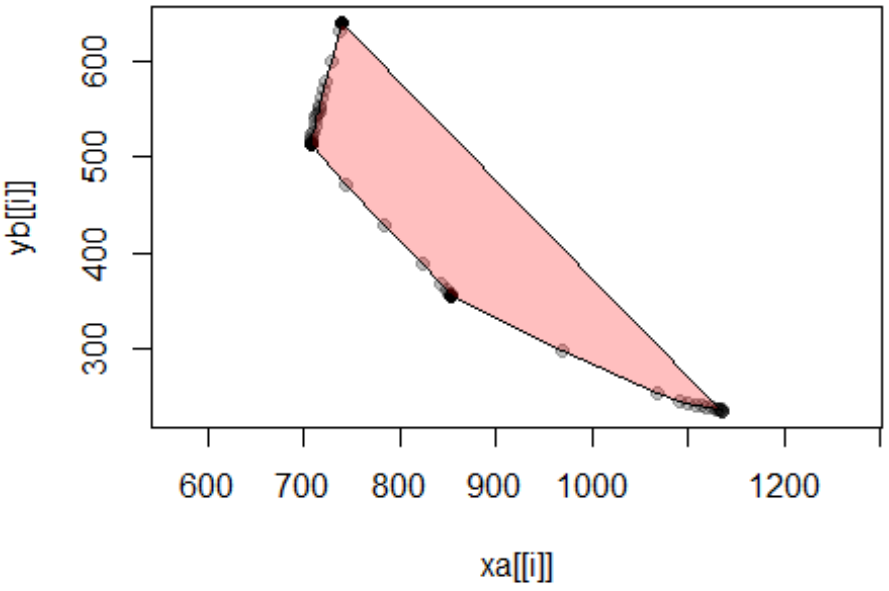

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

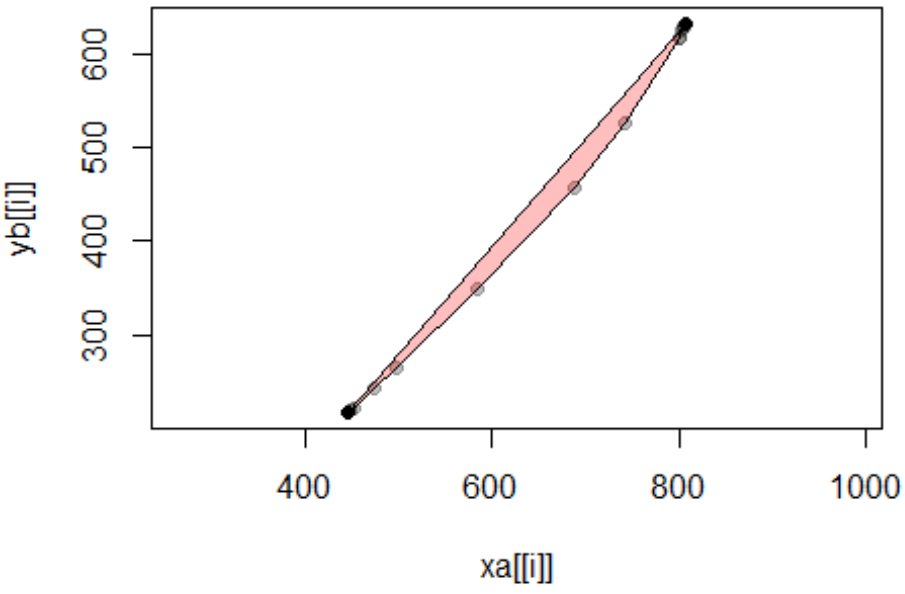

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

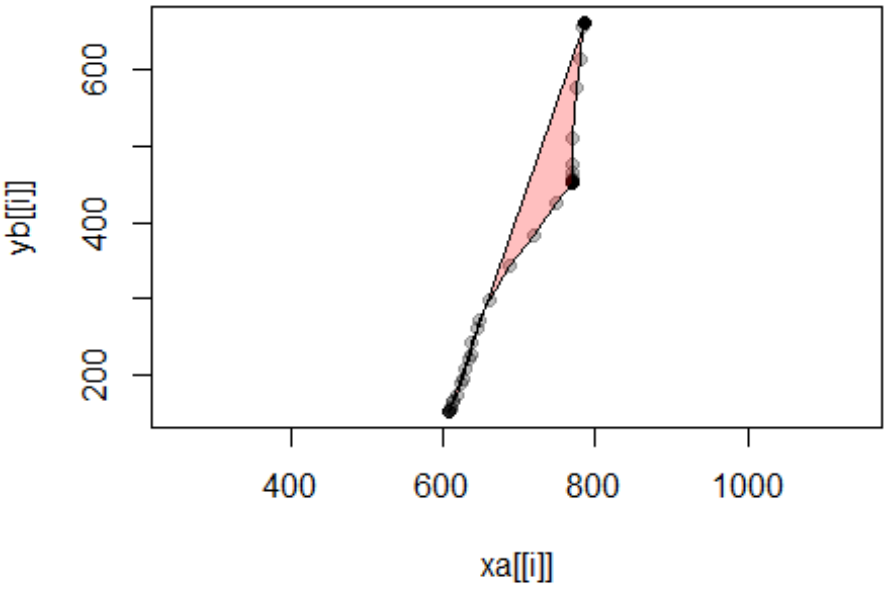

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

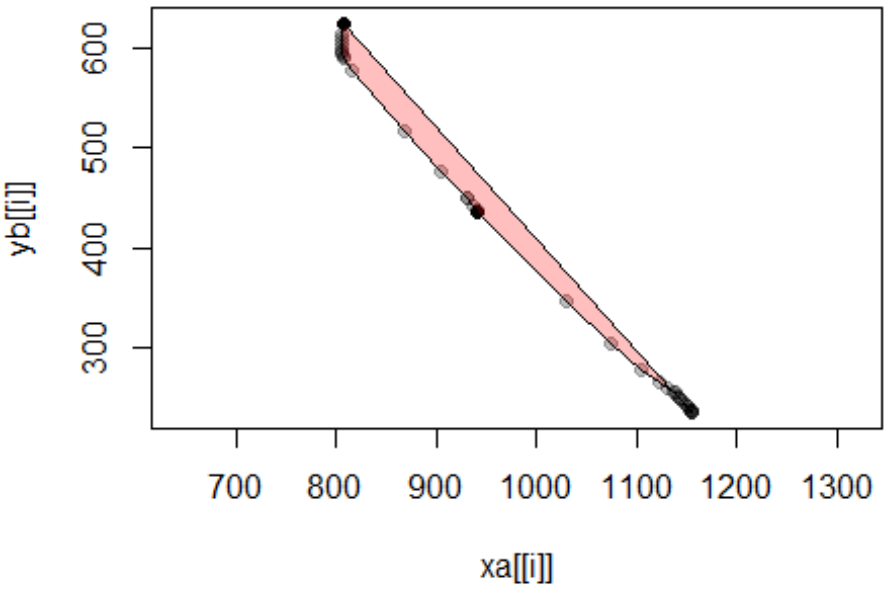

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

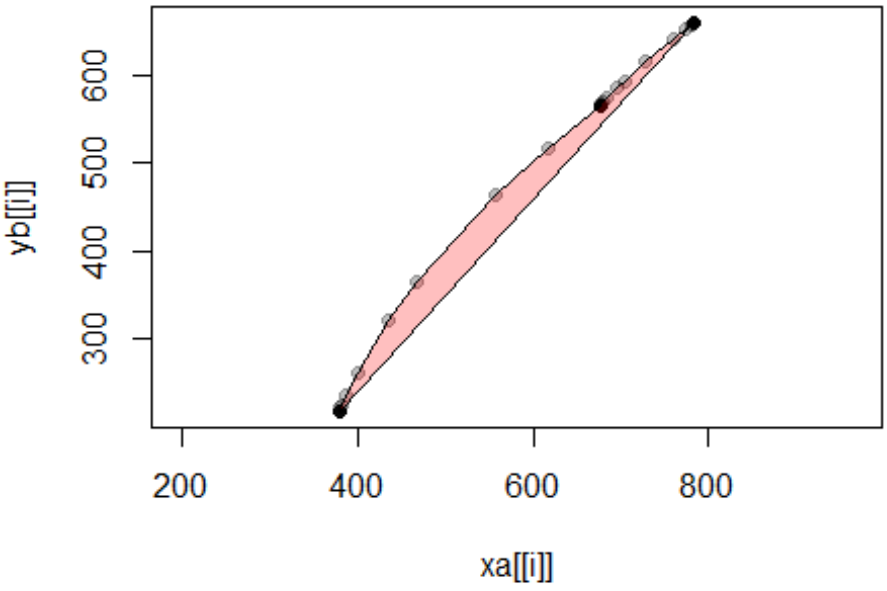

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

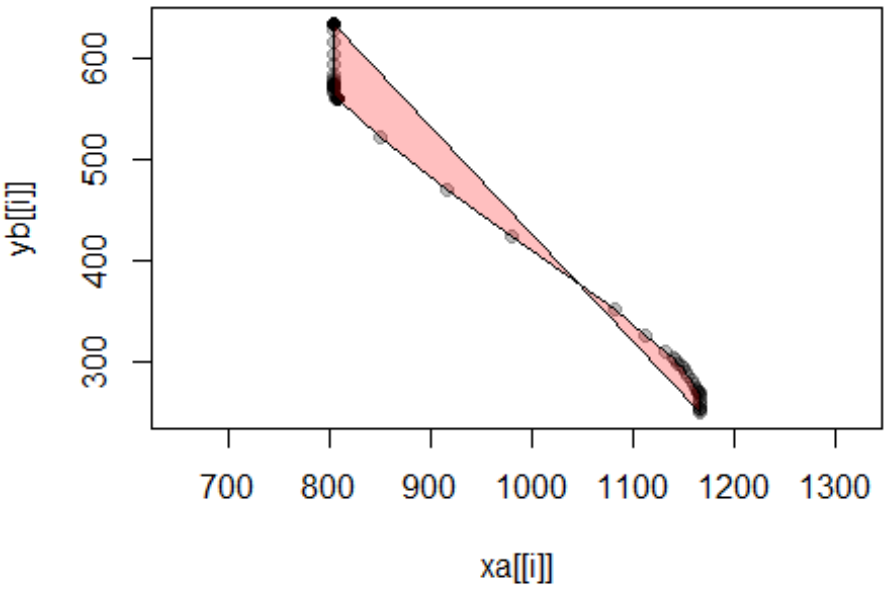

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

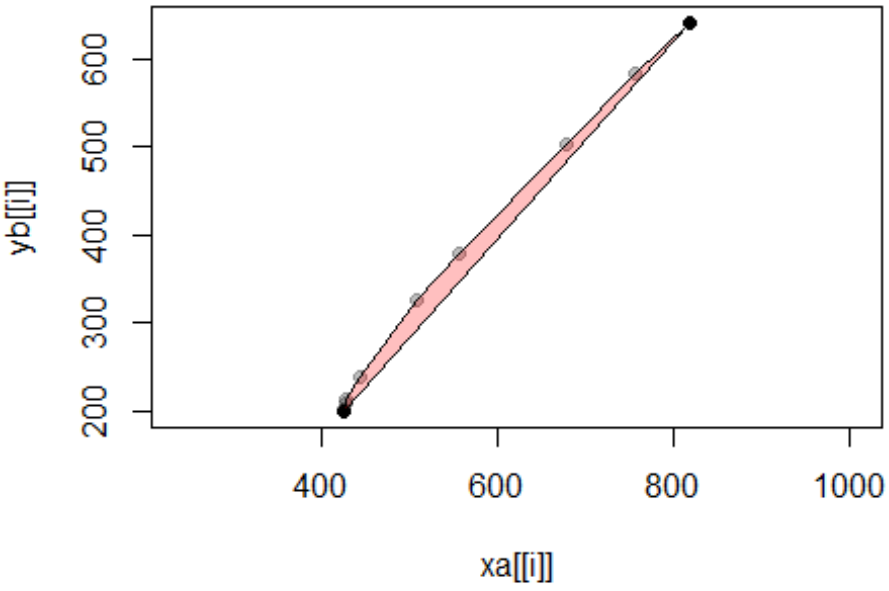

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

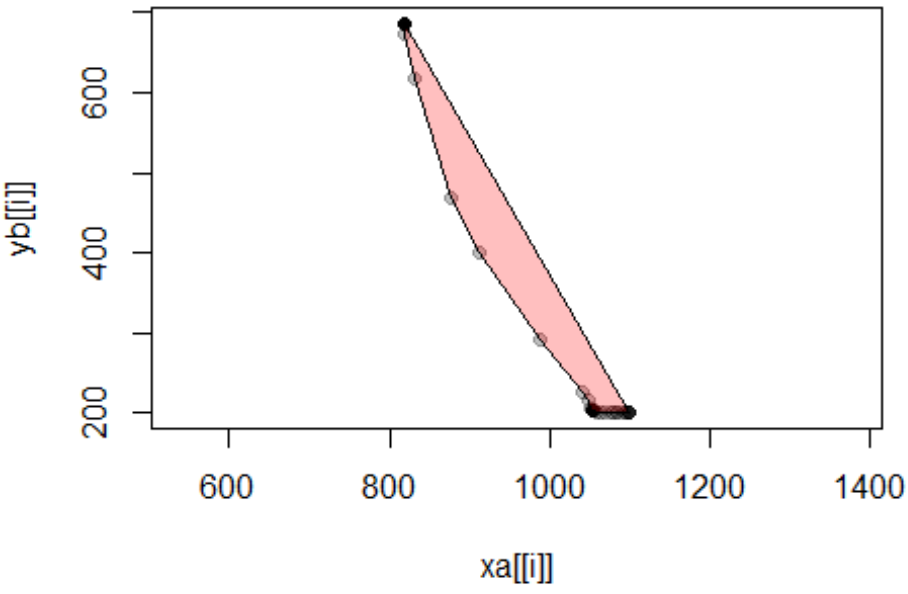

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

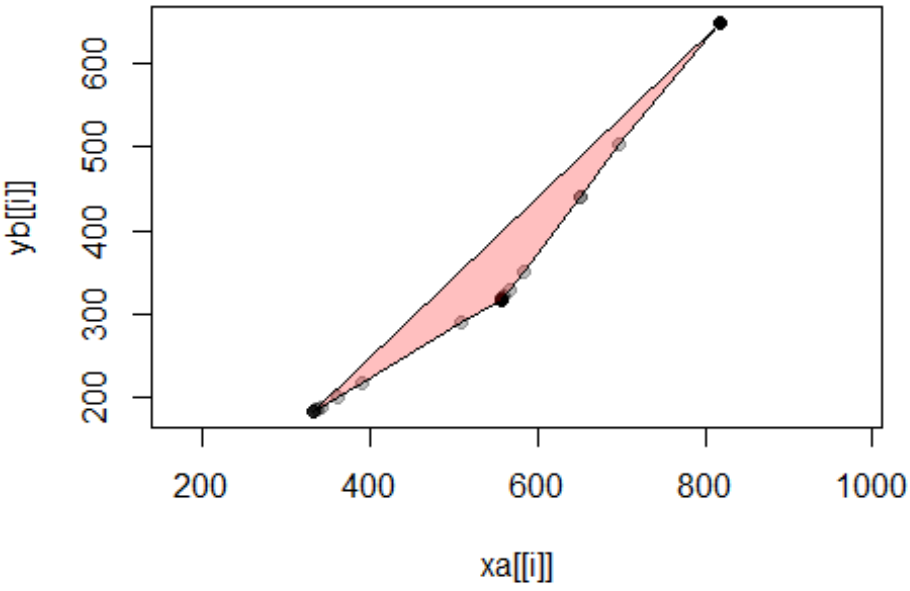

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

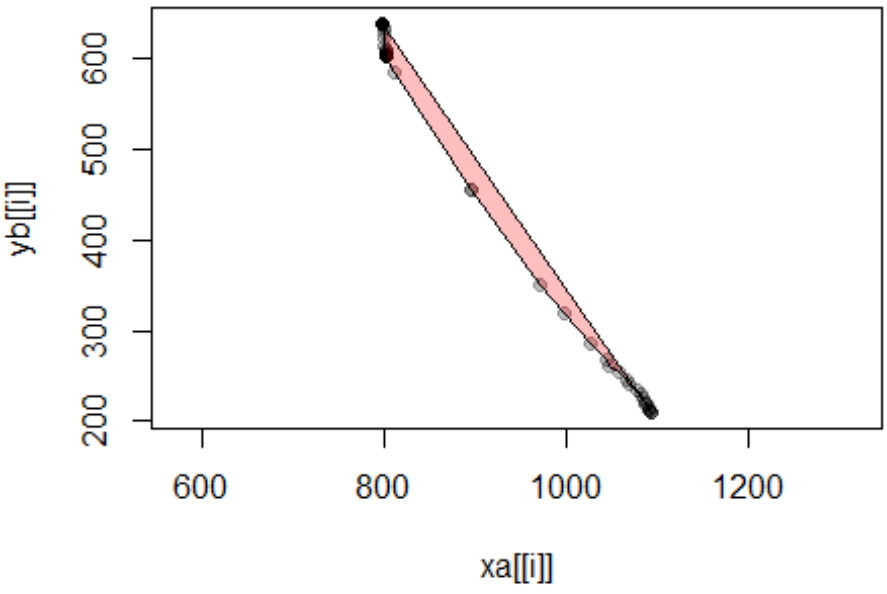

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

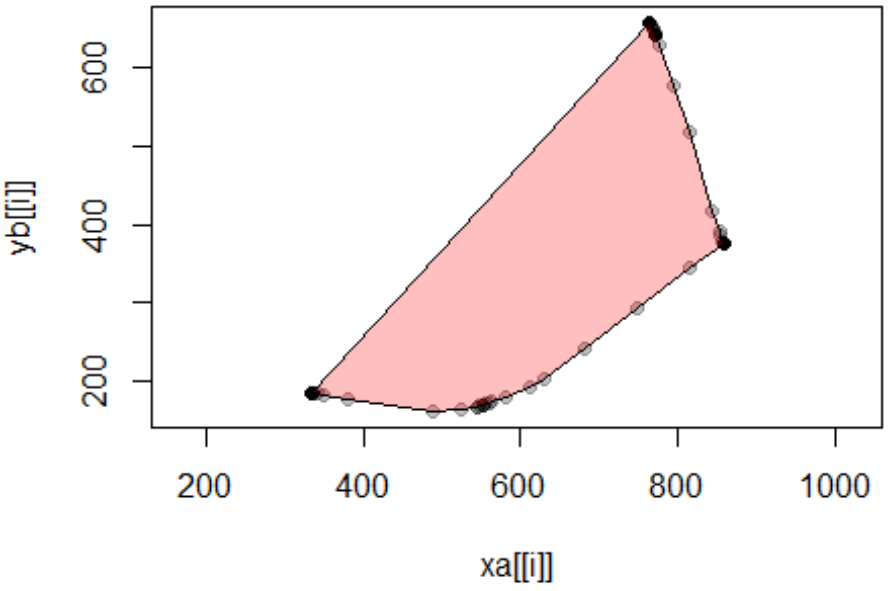

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

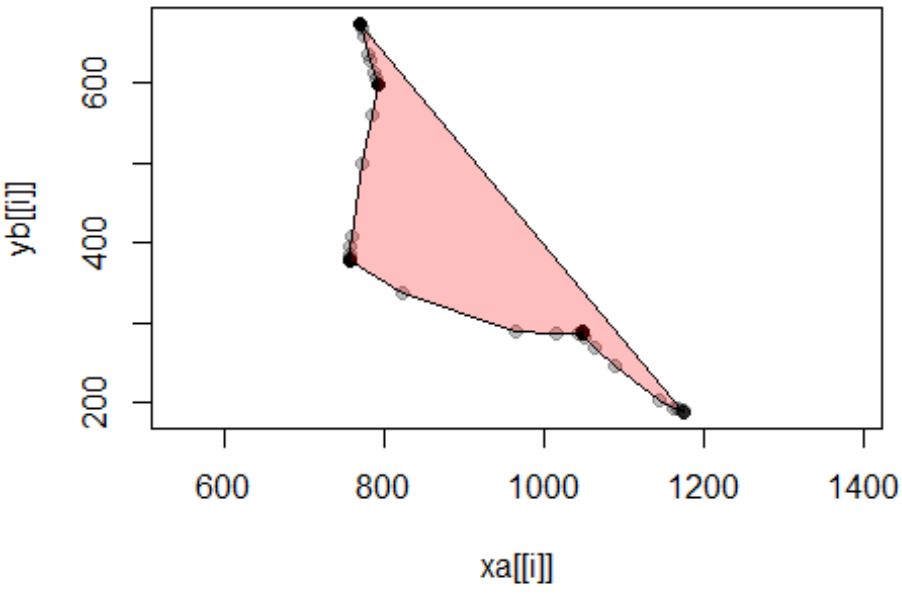

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

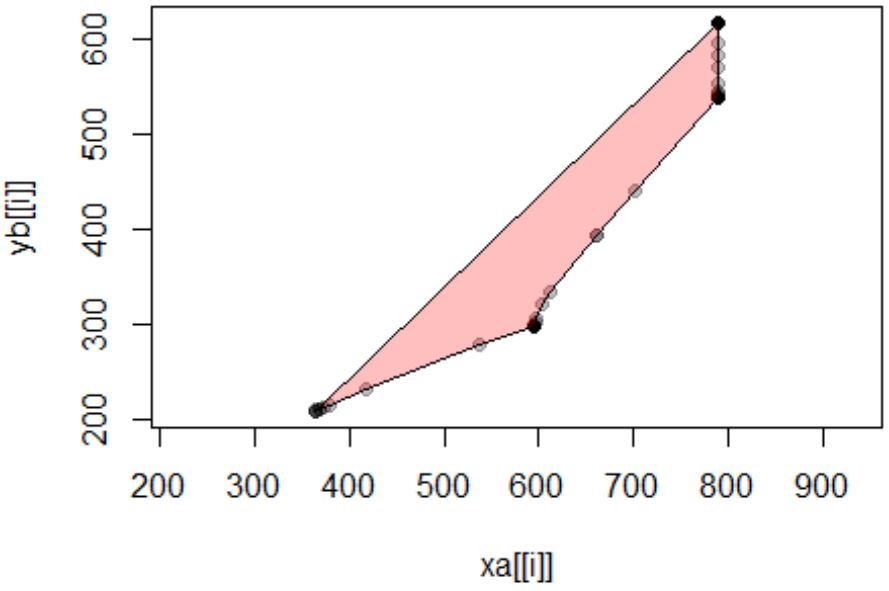

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

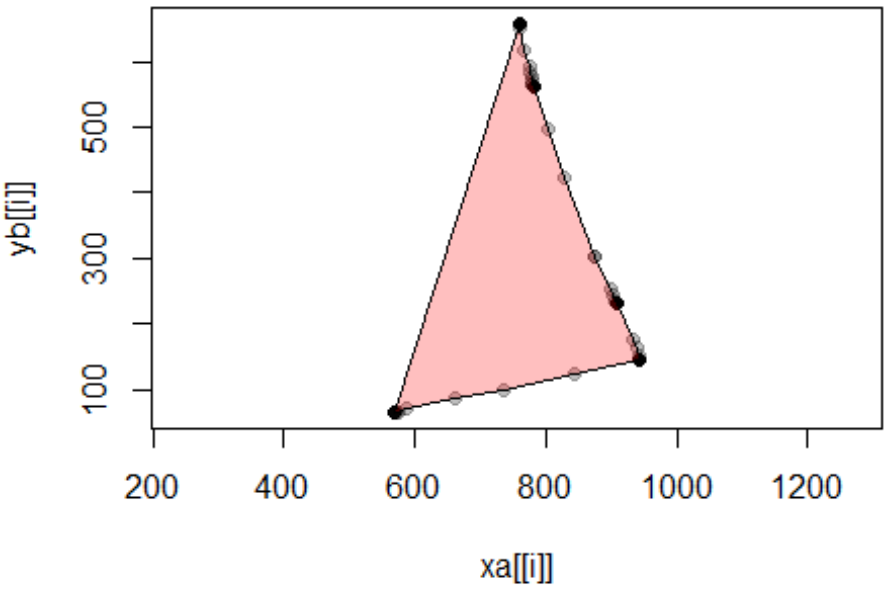

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

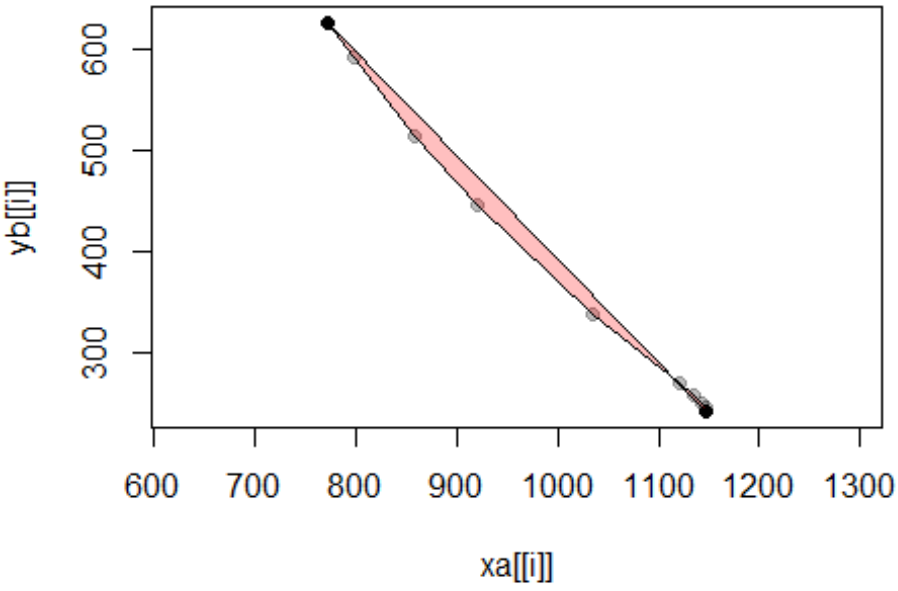

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

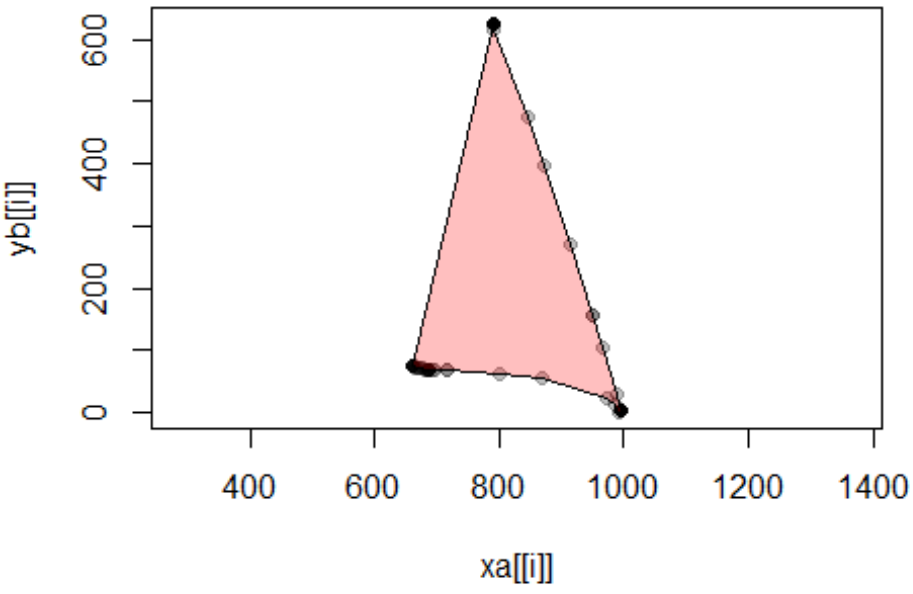

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

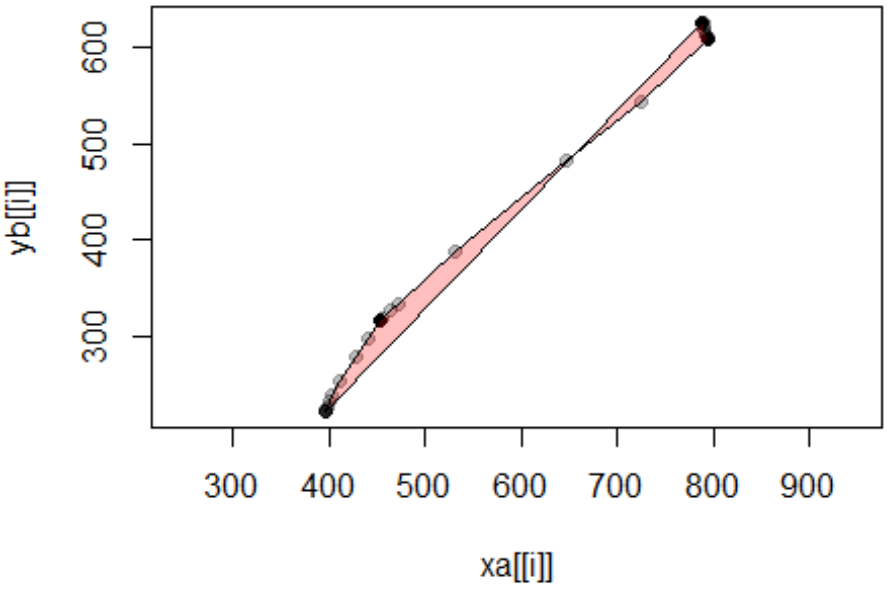

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

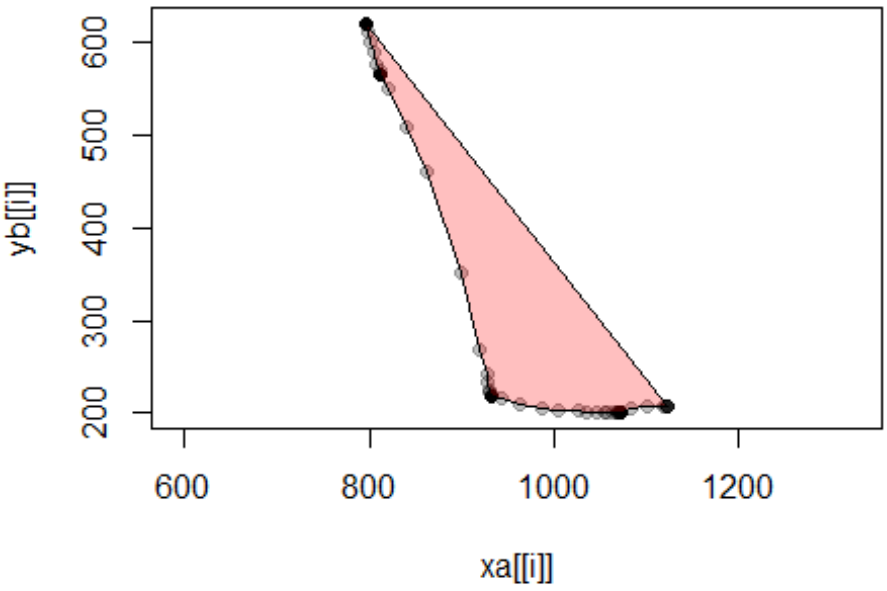

yb[[i]]
200
300
400
500
600
200
300
400
500
600
700
800
900
xa[[i]]

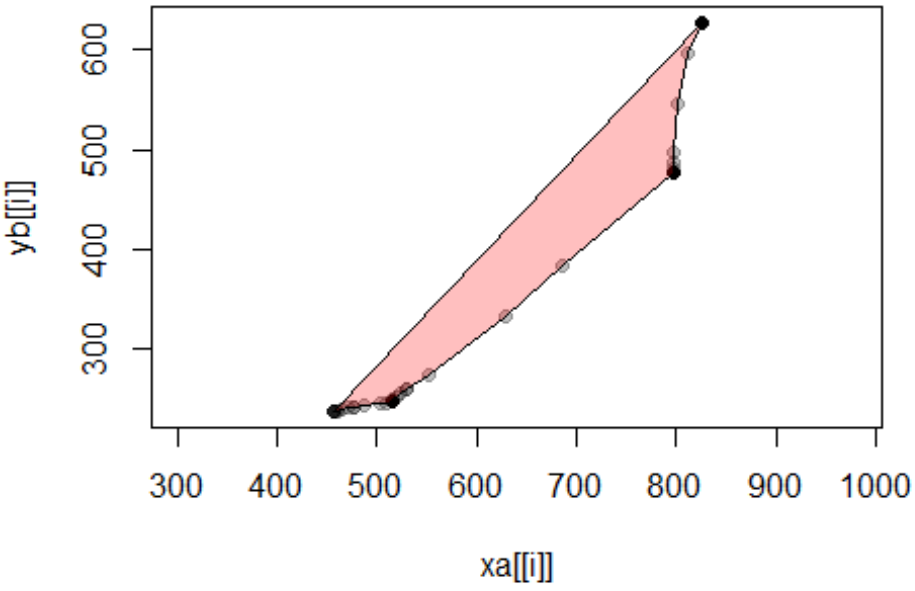

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

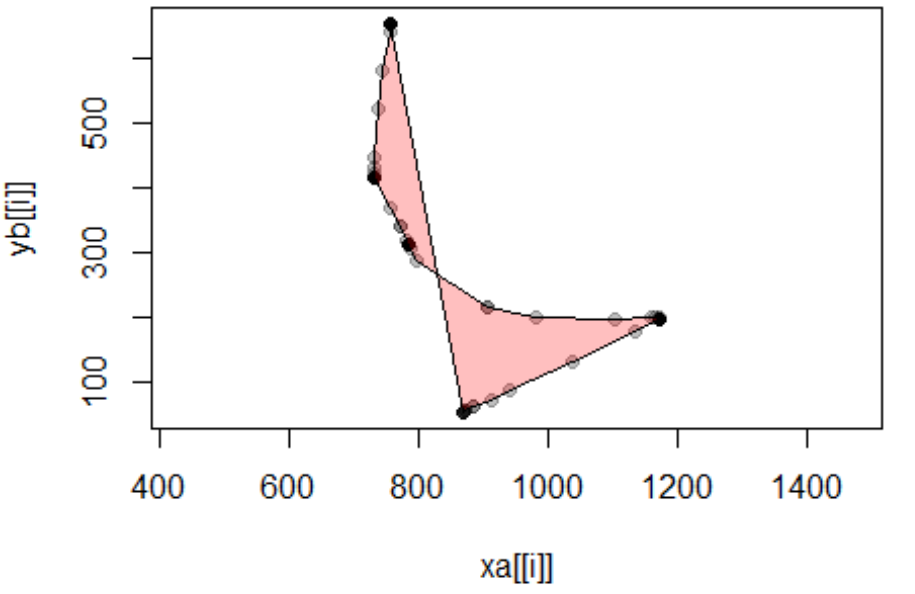

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
1300
xa[[i]]

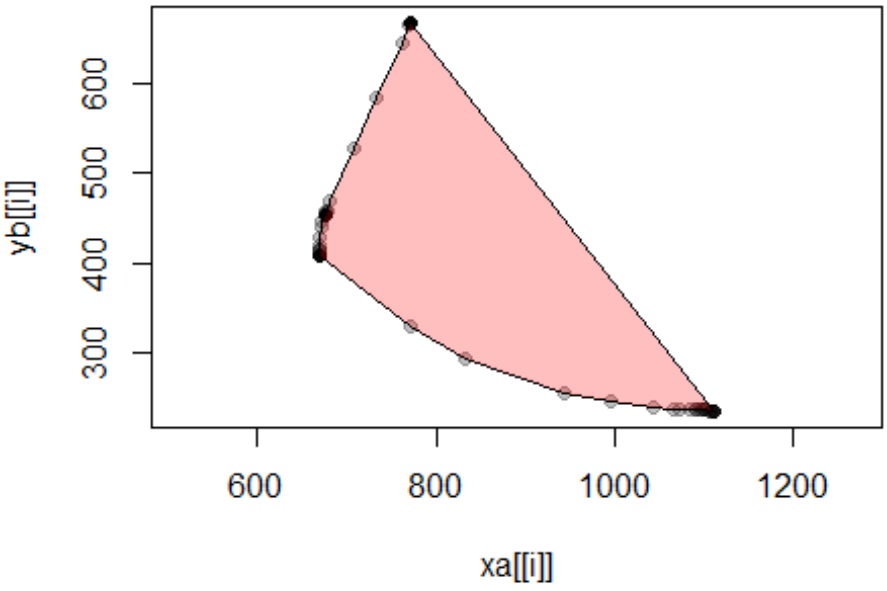

yb[[i]]
0
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

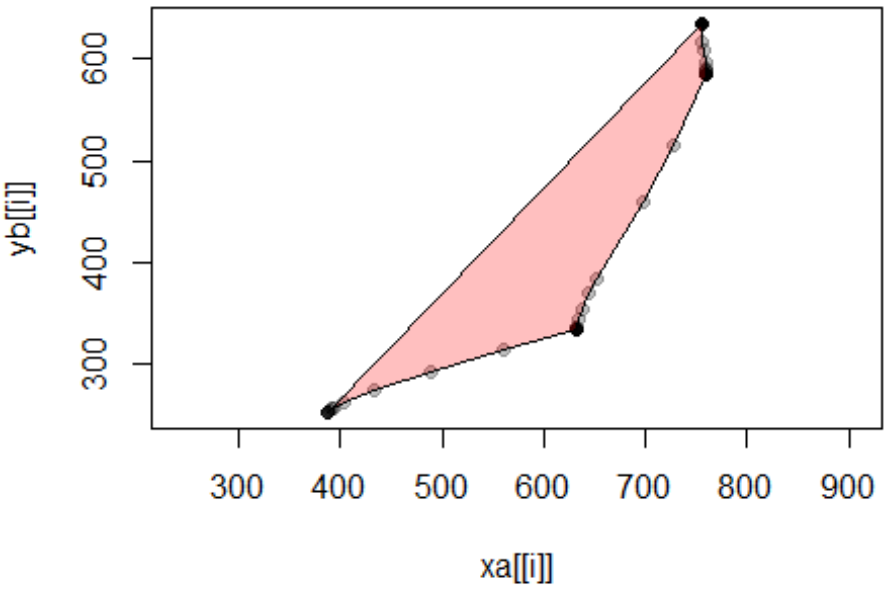

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

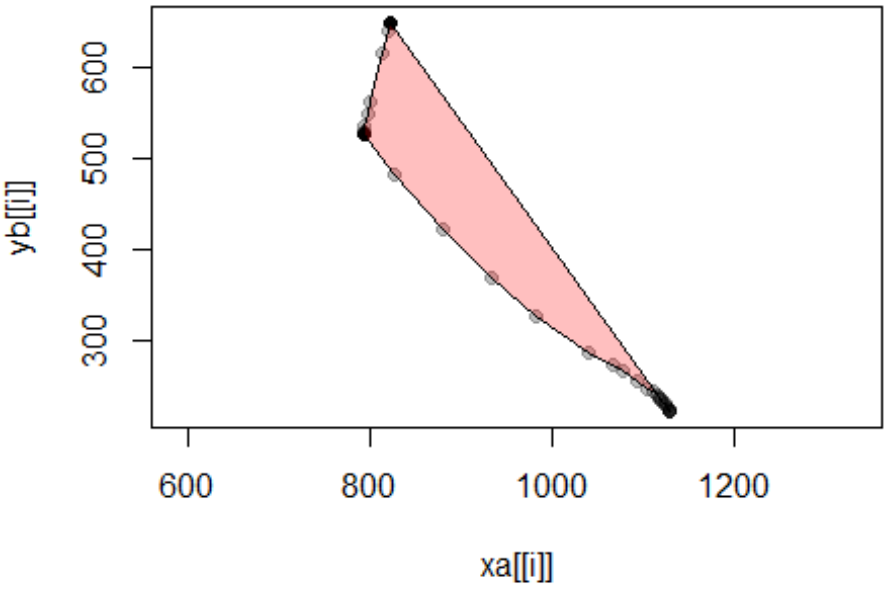

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

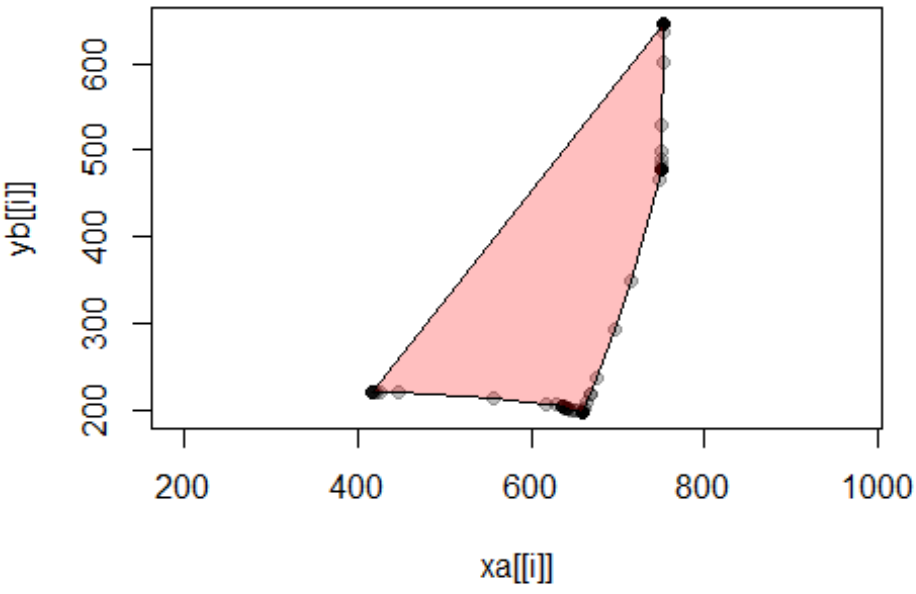

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
1000
xa[[i]]

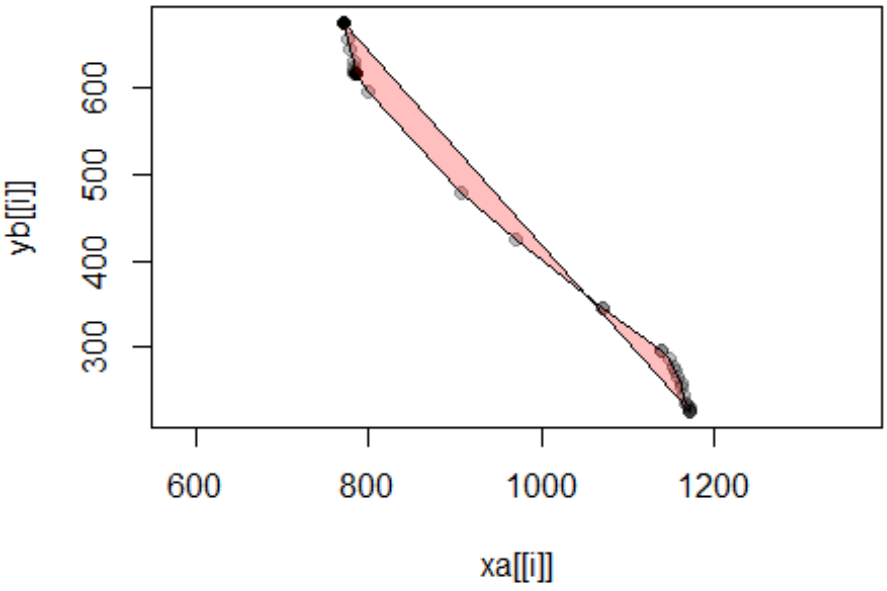

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

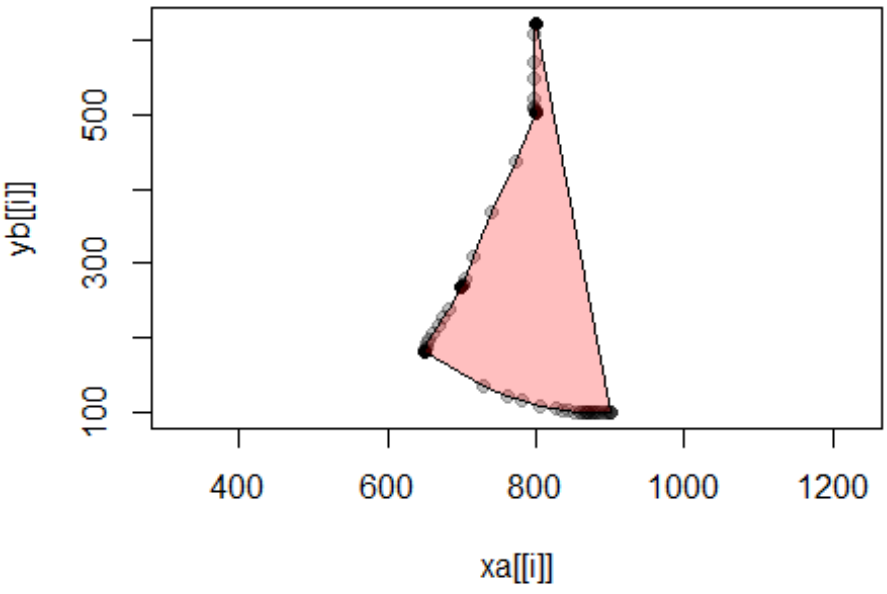

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

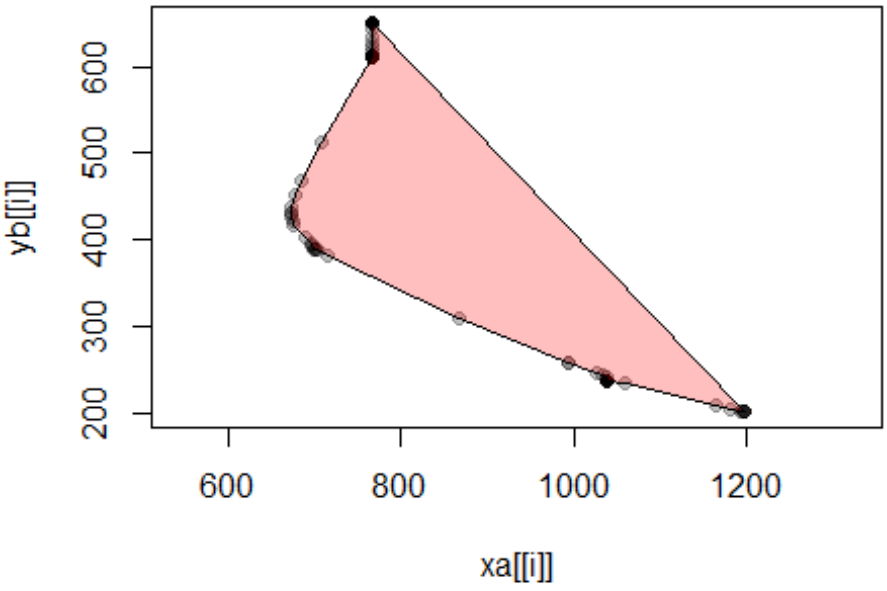

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

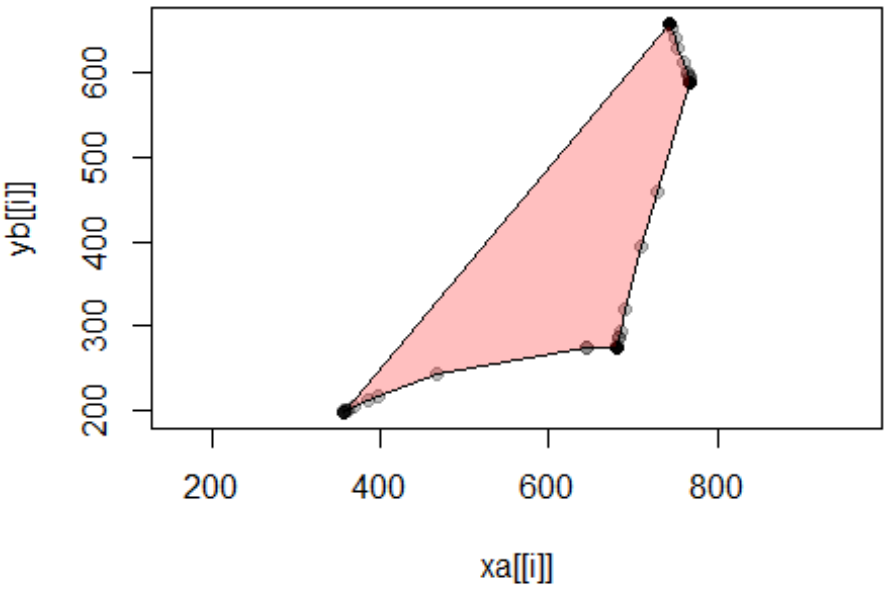

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

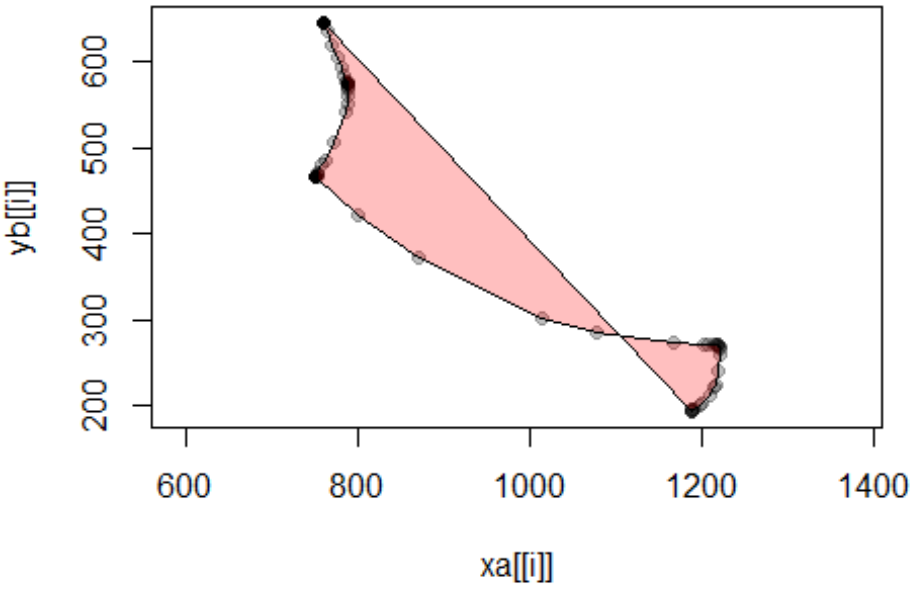

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

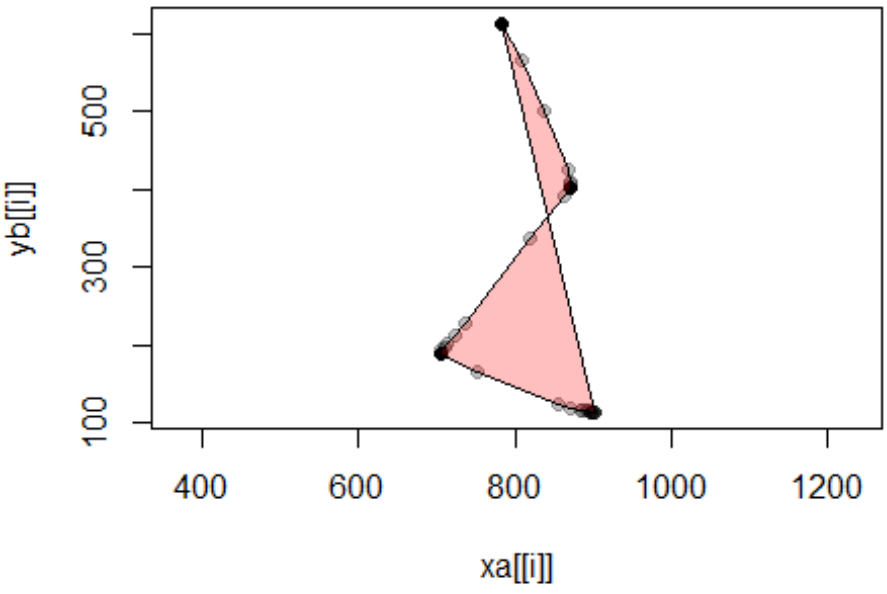

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

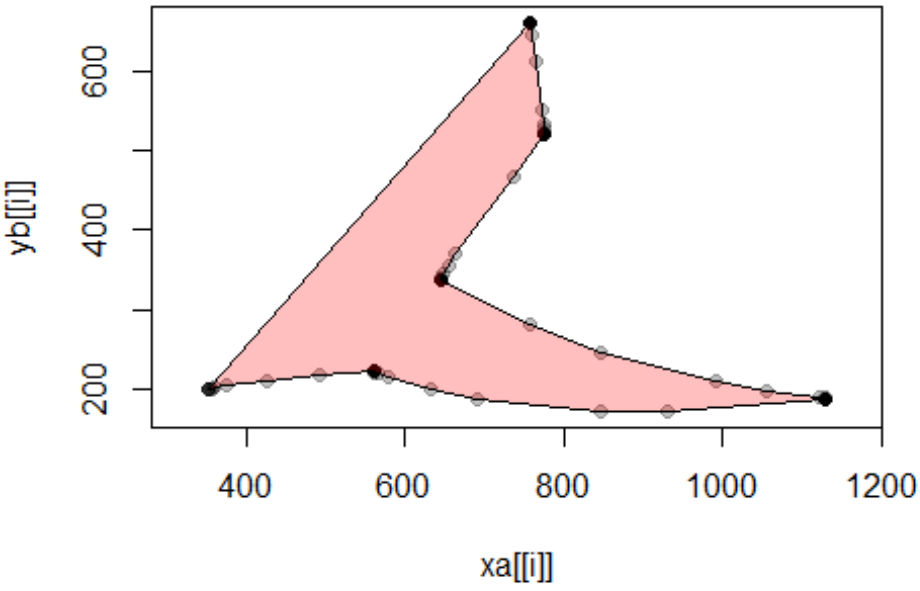

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

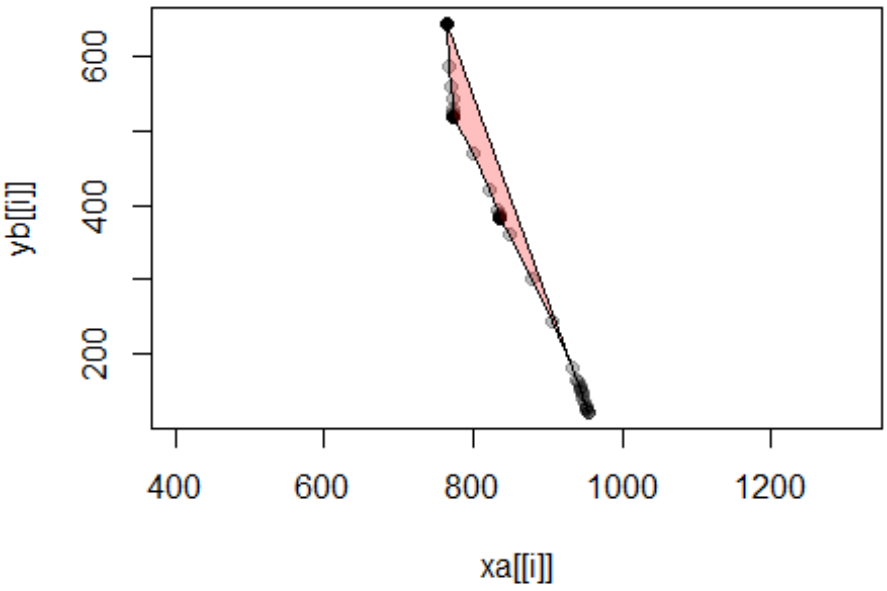

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

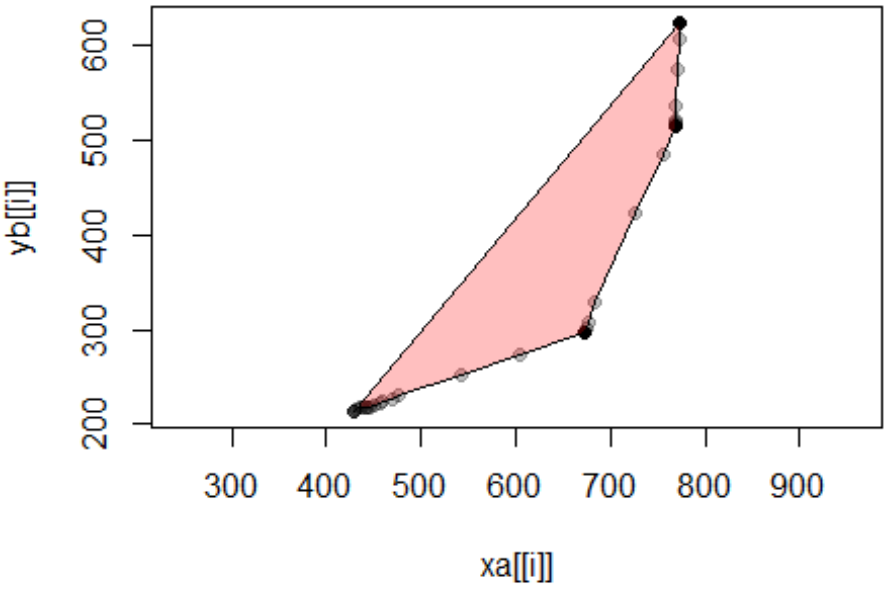

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

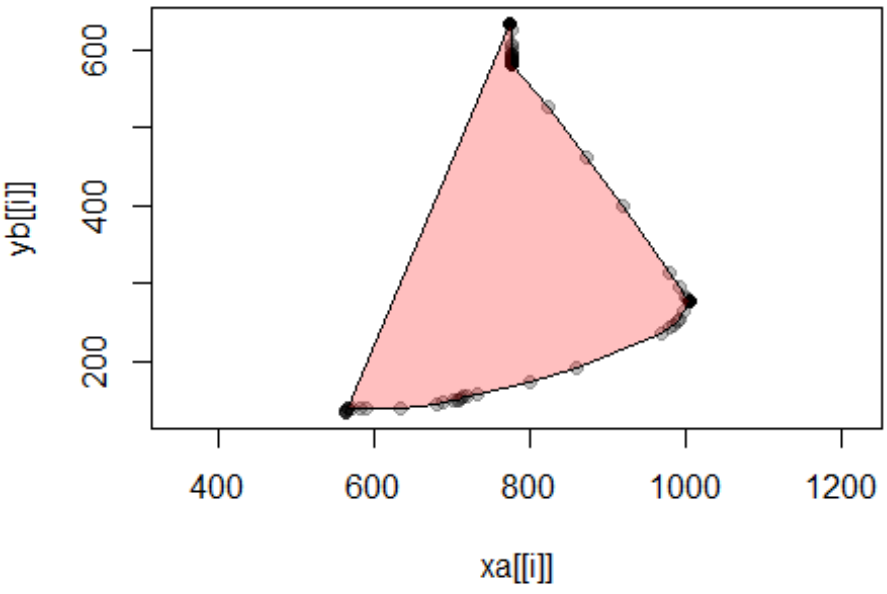

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

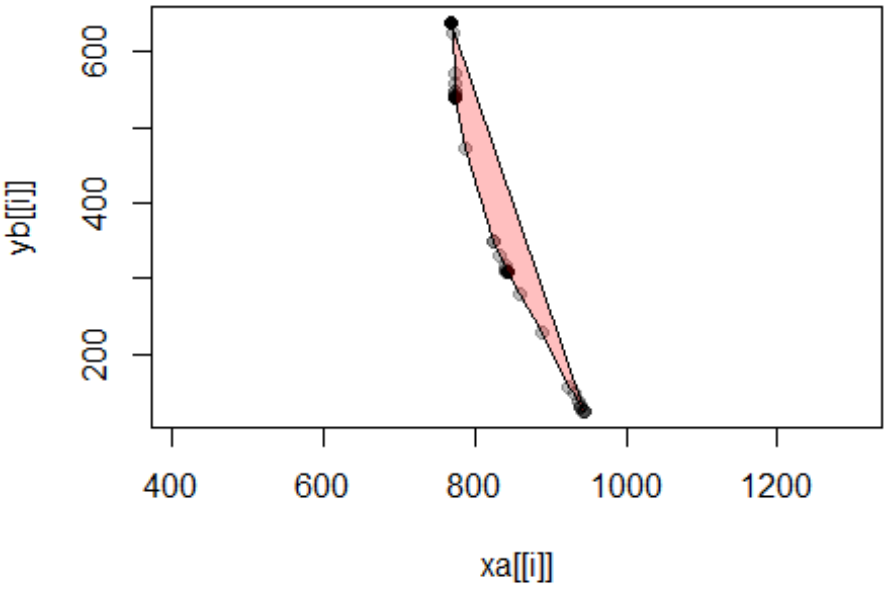

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

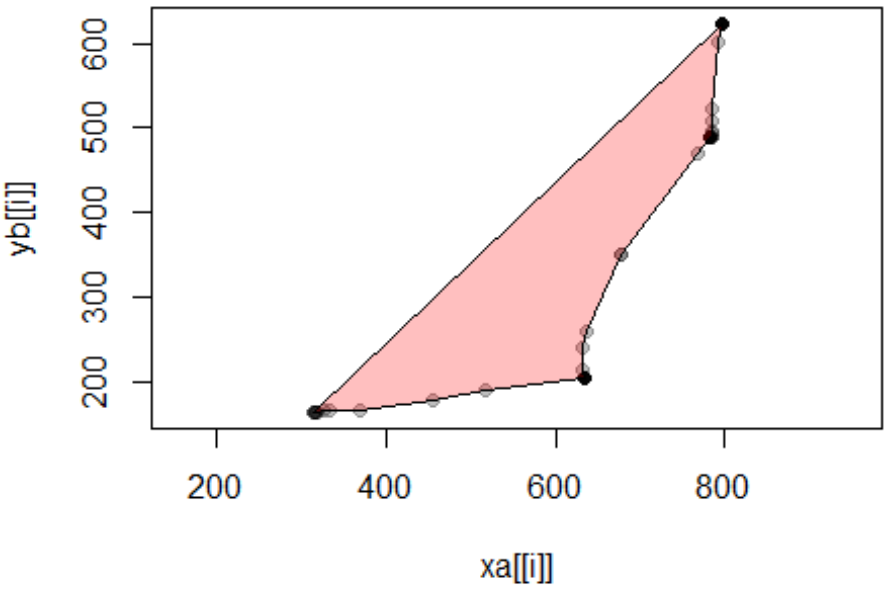

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

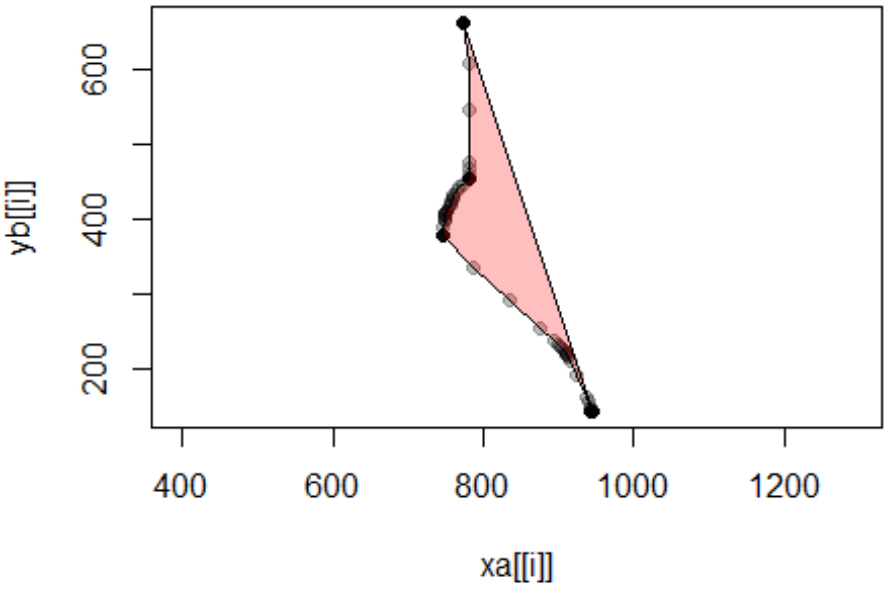

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

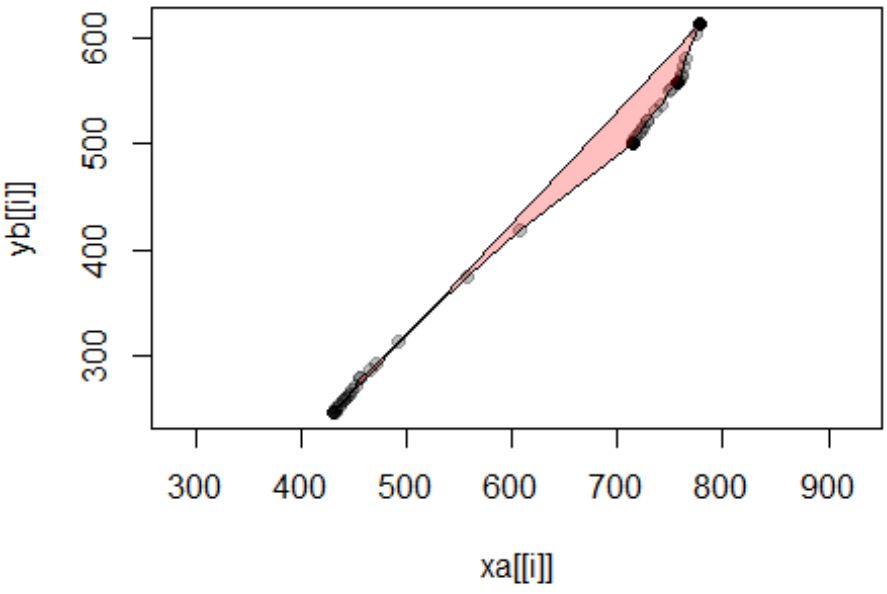

yb[[i]]
200
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

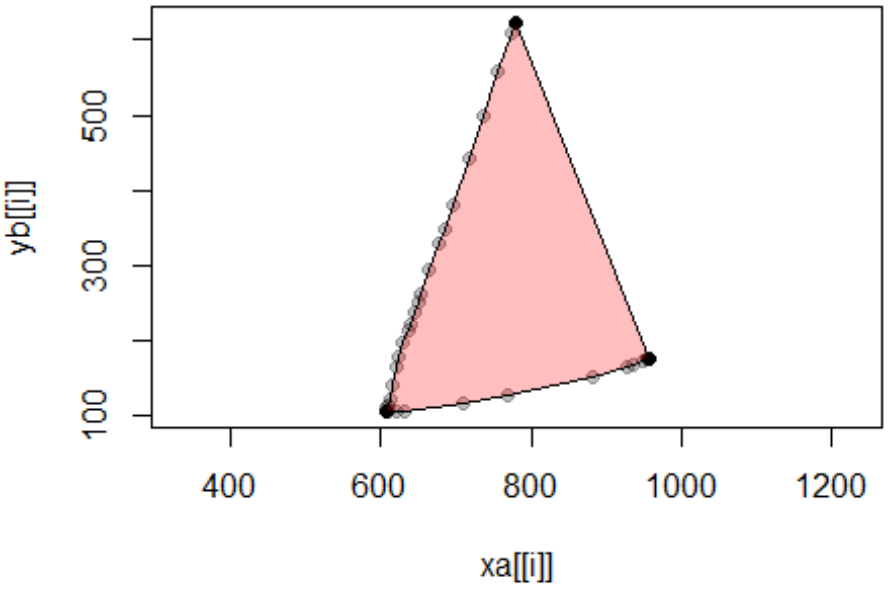

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

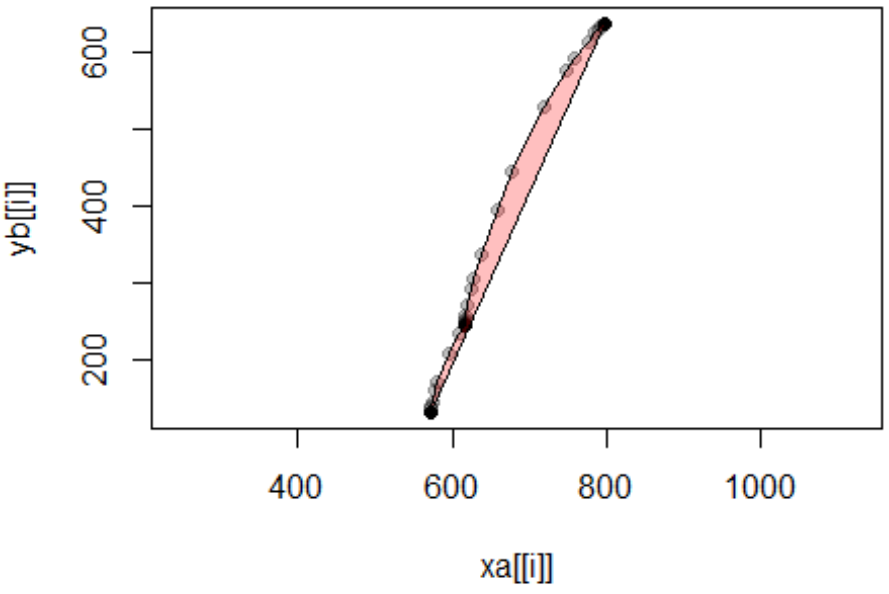

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

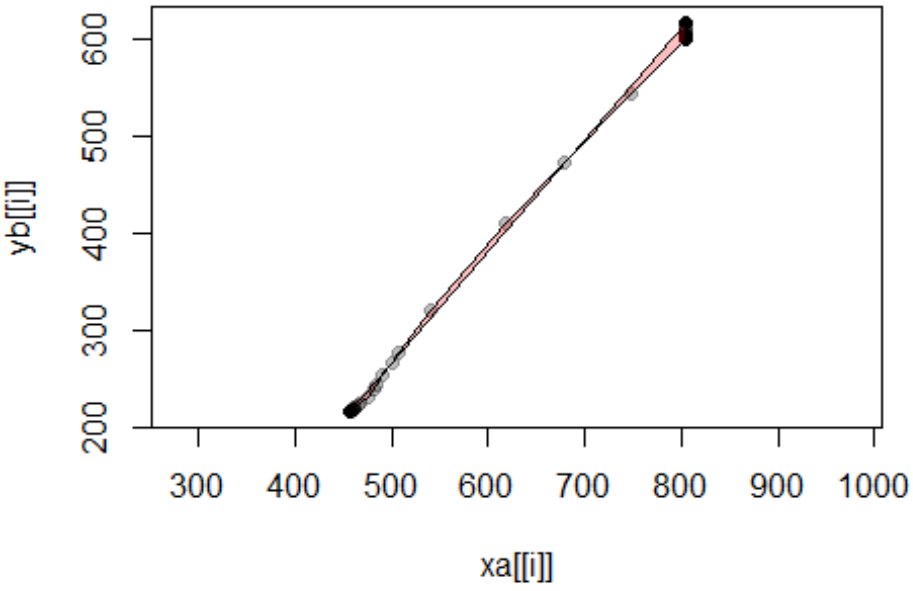

yb[[i]]
600
500
400
300
200
200
400
600
800
xa[[i]]

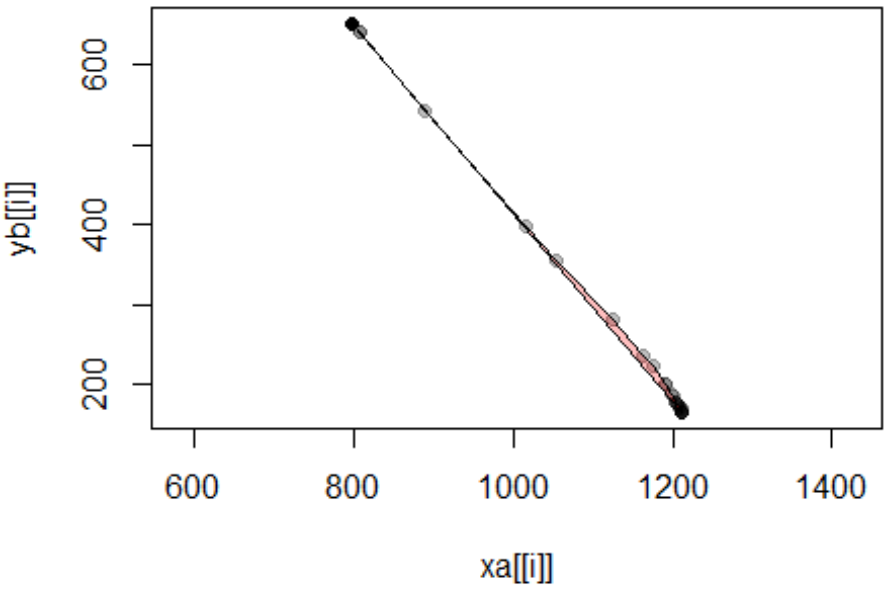

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

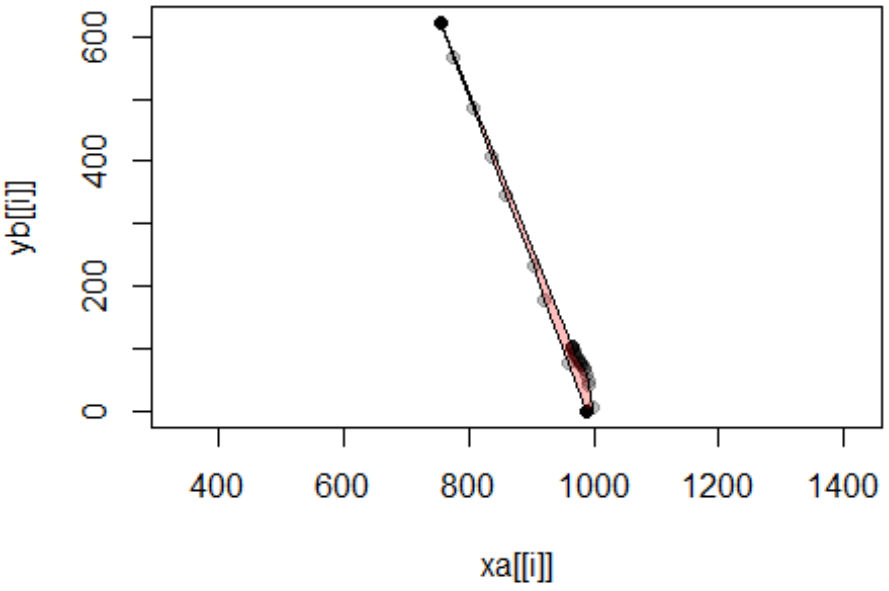

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

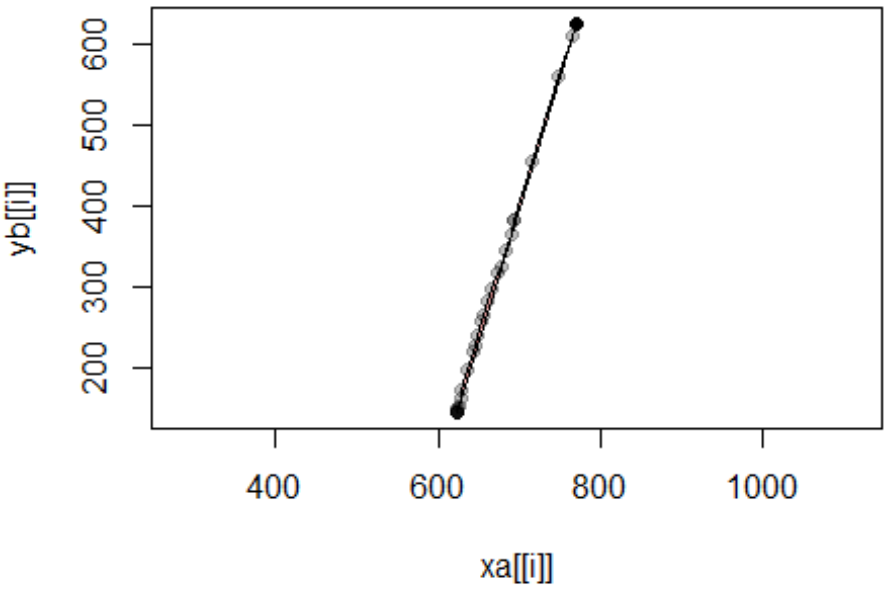

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

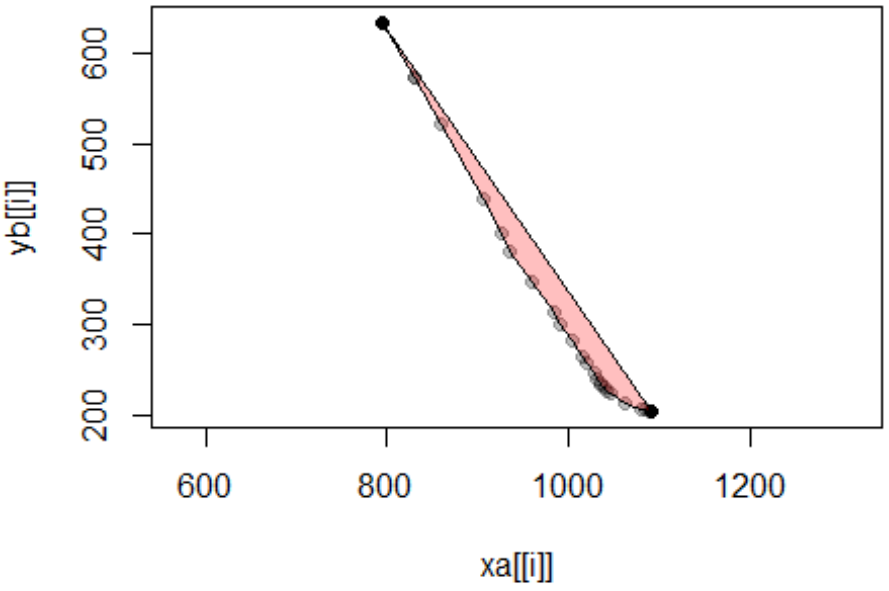

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

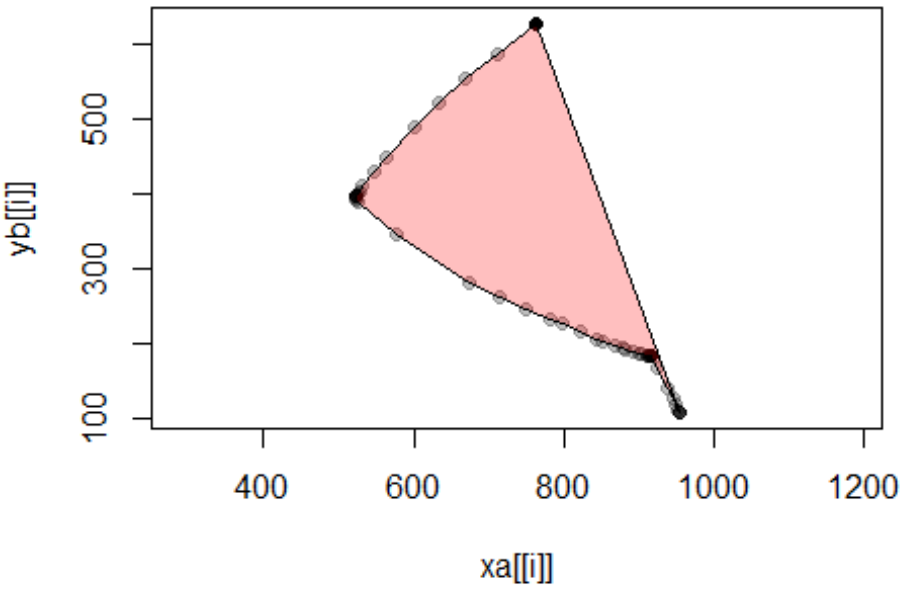

yb[[i]]
200
300
400
500
600
300
400
500
600
700
800
900
1000
xa[[i]]

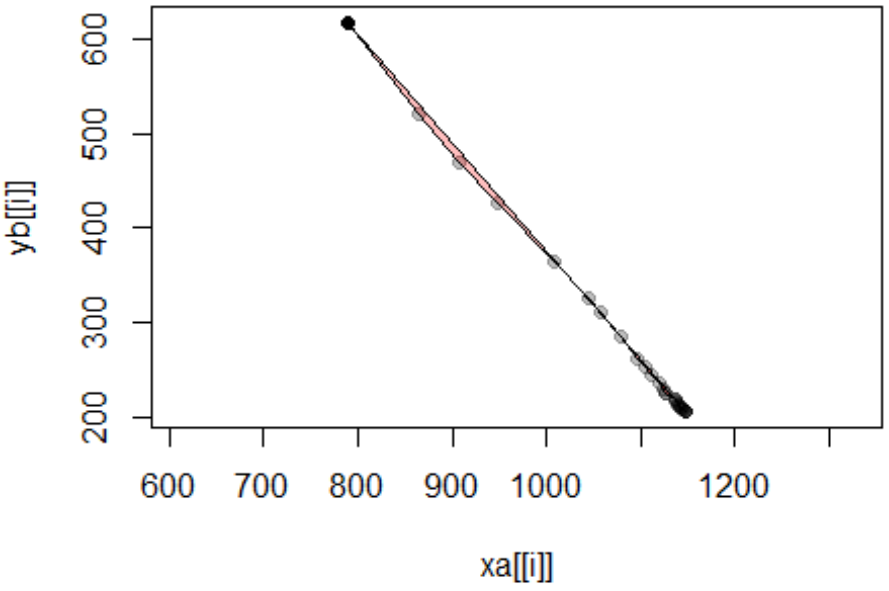

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

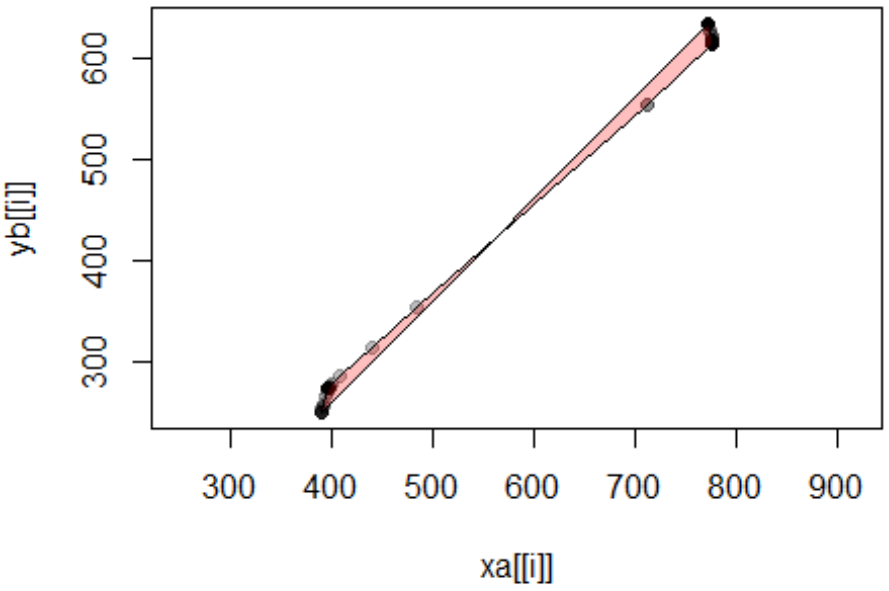

yb[[i]]
0
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

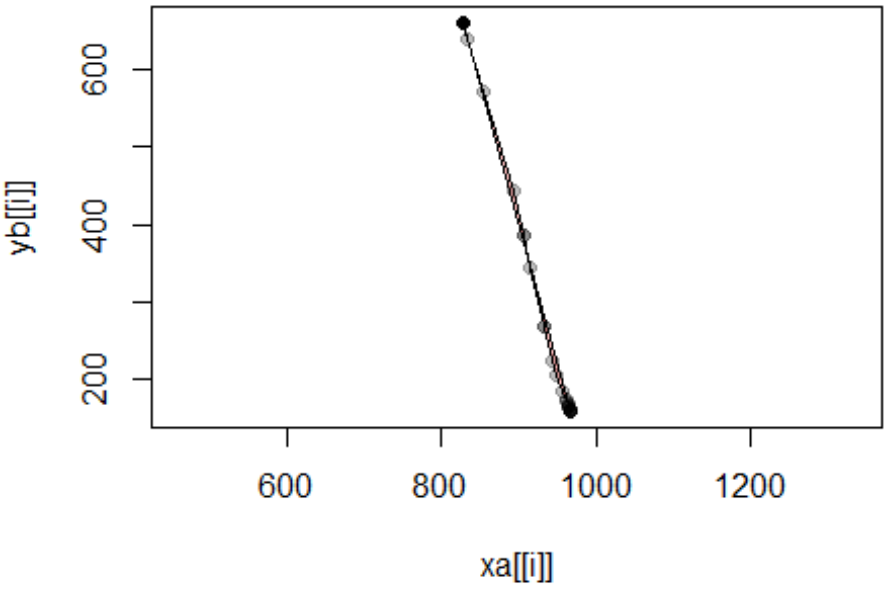

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

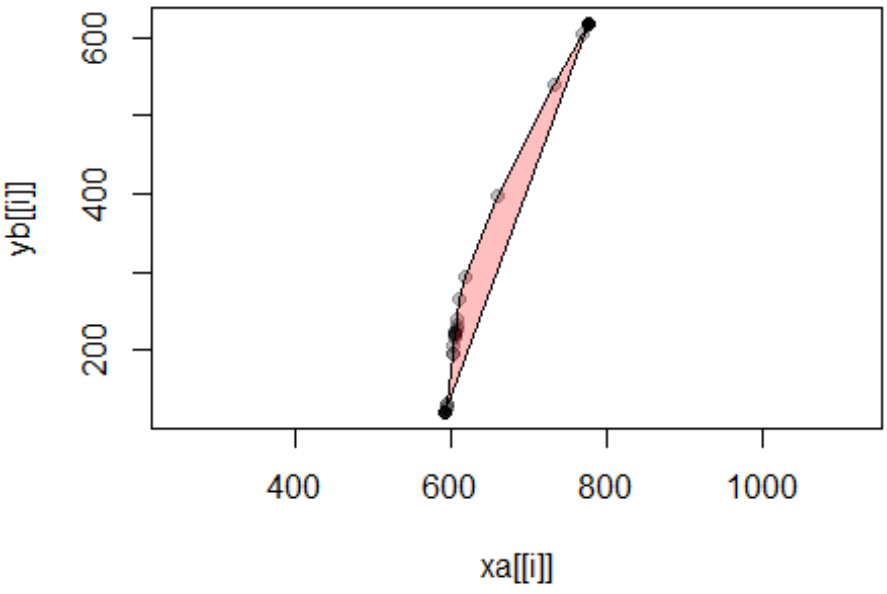

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

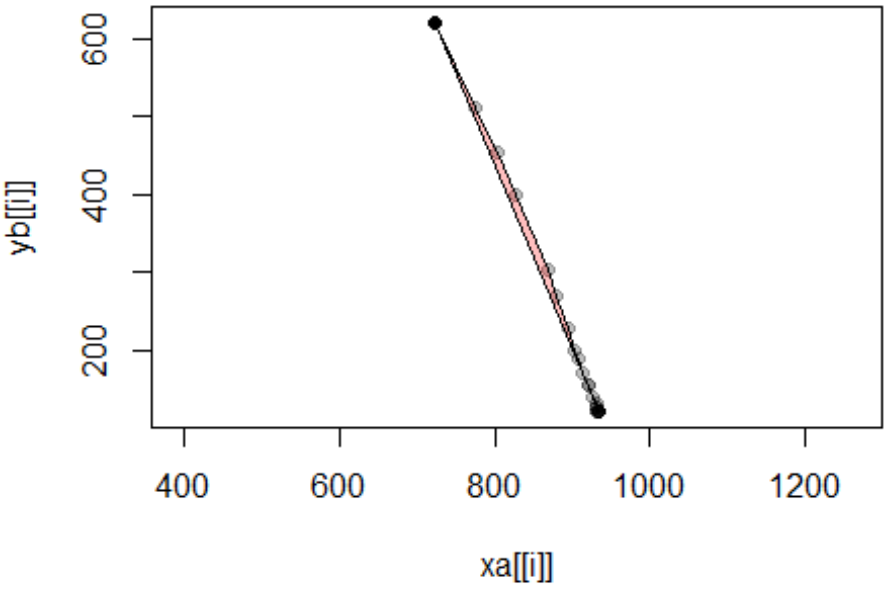

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

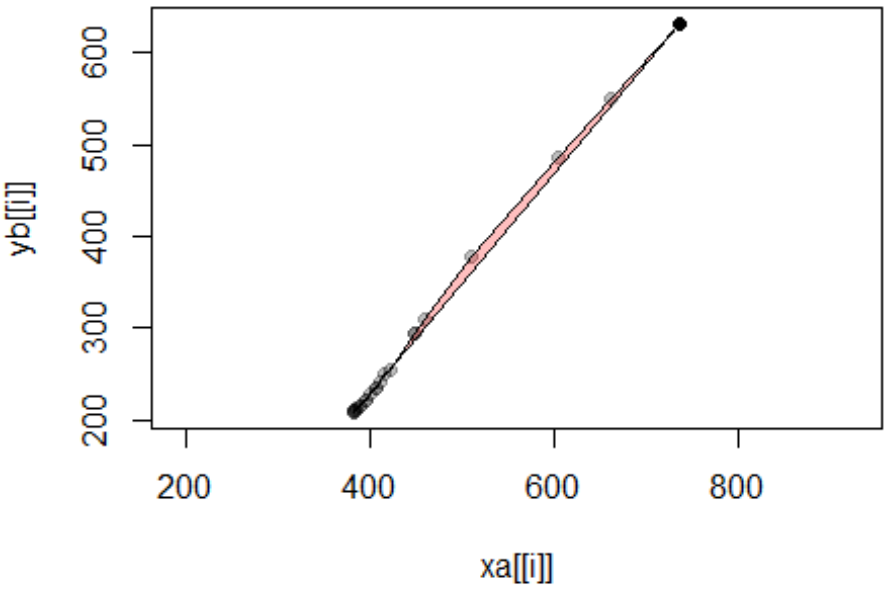

yb[[i]]
200
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

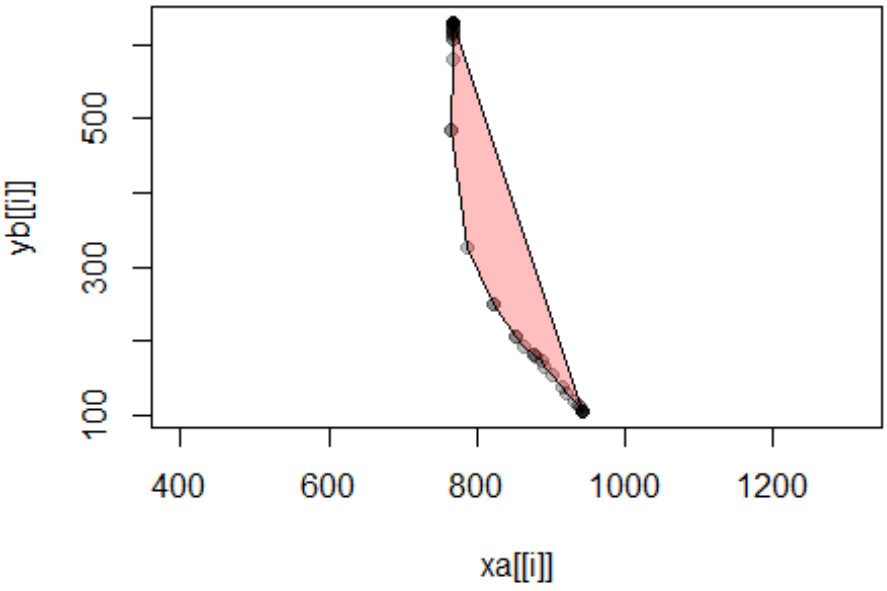

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

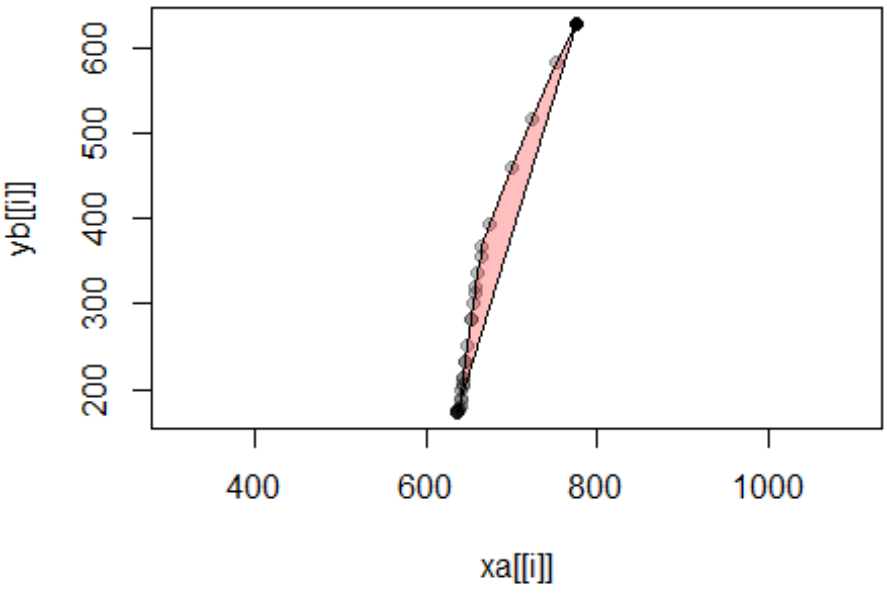

yb[[i]]
600
400
200
600
800
1000
1200
xa[[i]]

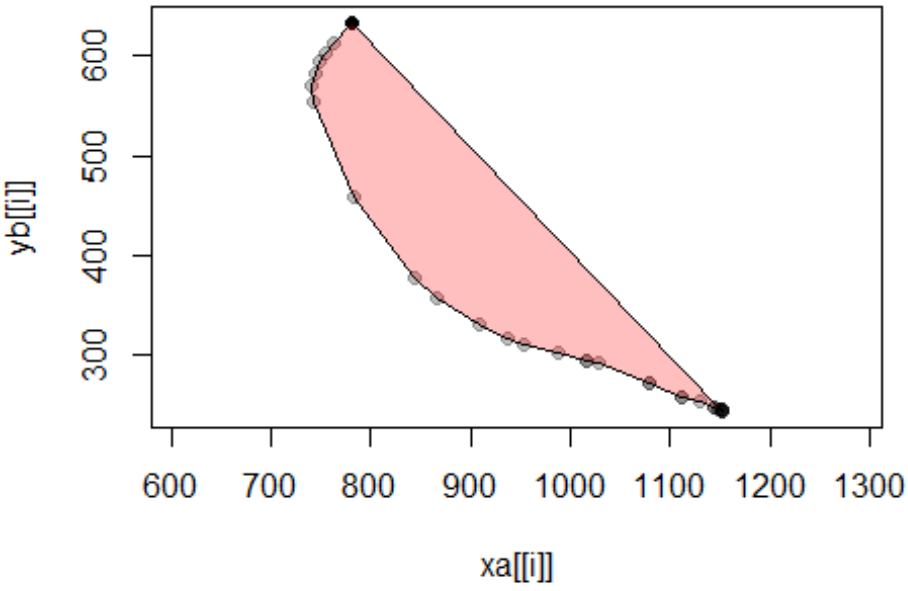

yb[[i]]
600
400
200
400
600
800
1000
xa[[i]]

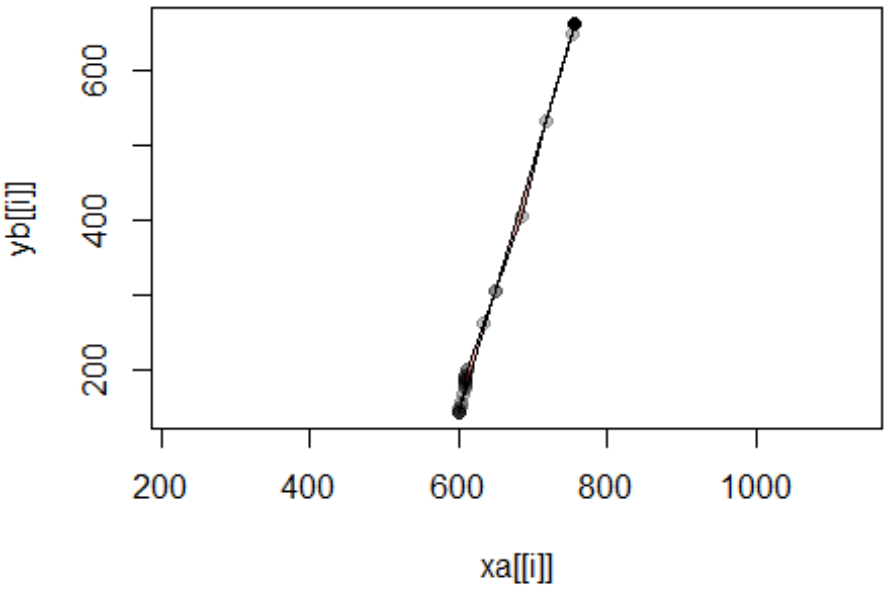

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

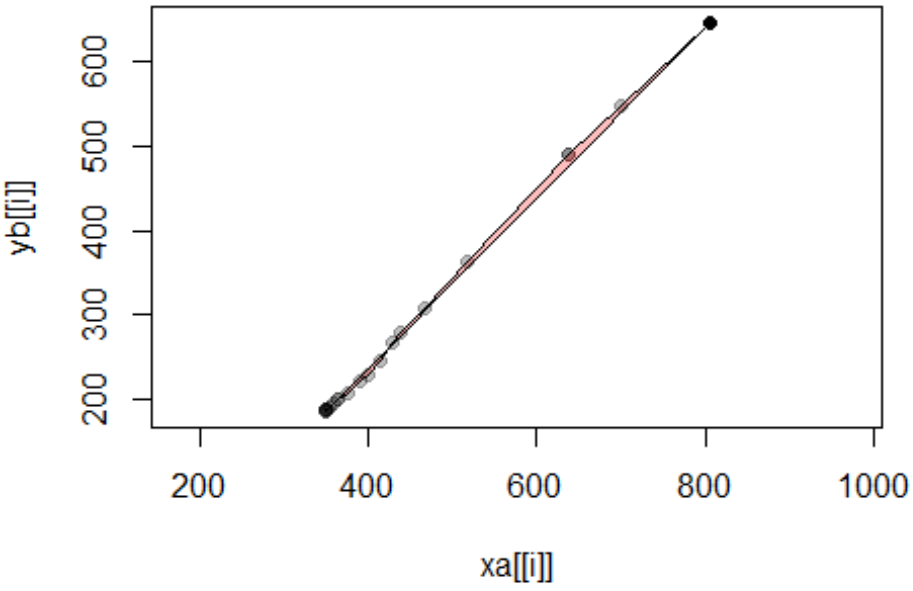

yb[[i]]
600
500
400
300
200
200
400
600
800
xa[[i]]

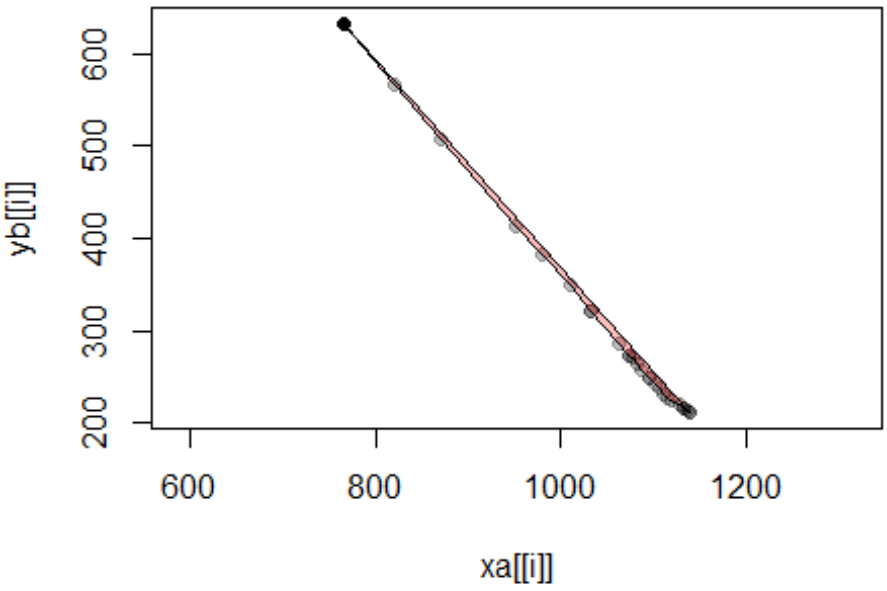

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

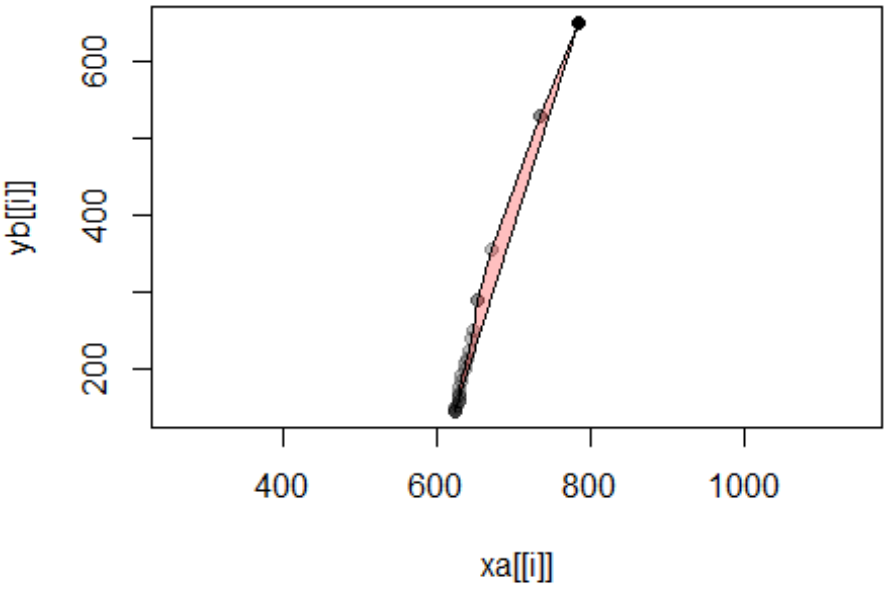

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

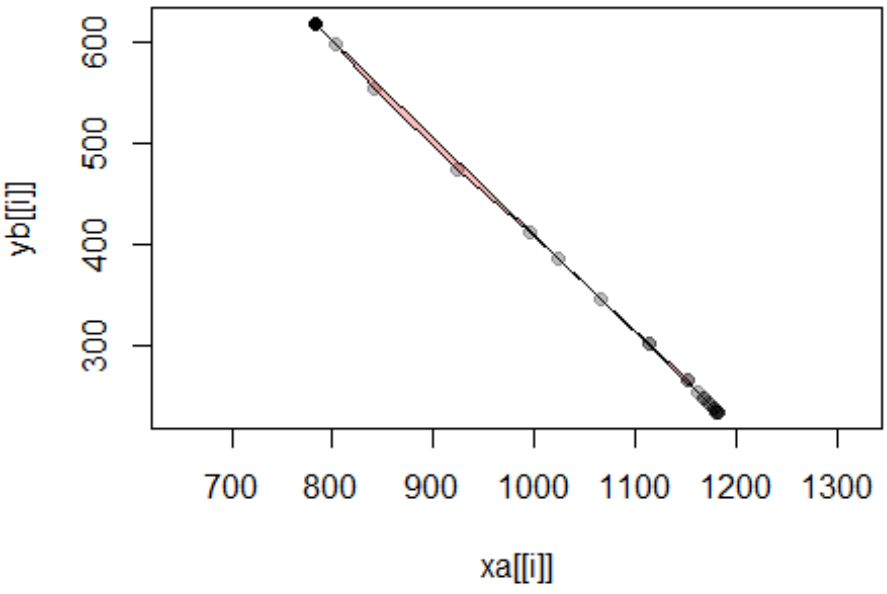

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
1300
xa[[i]]

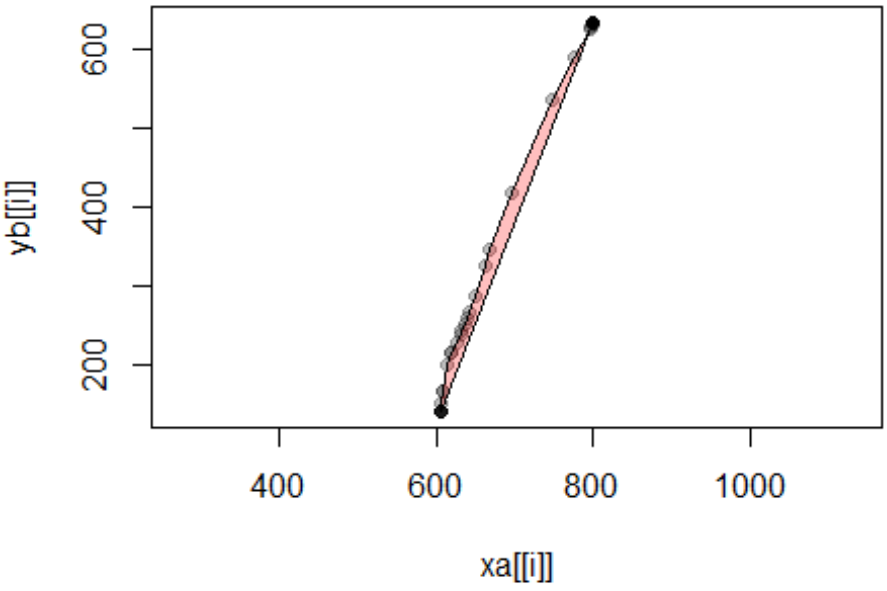

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

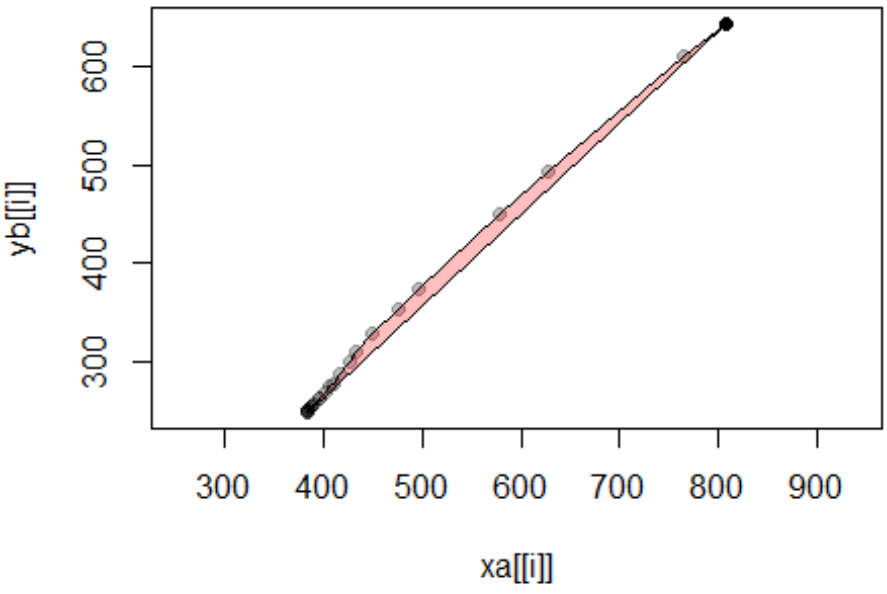

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

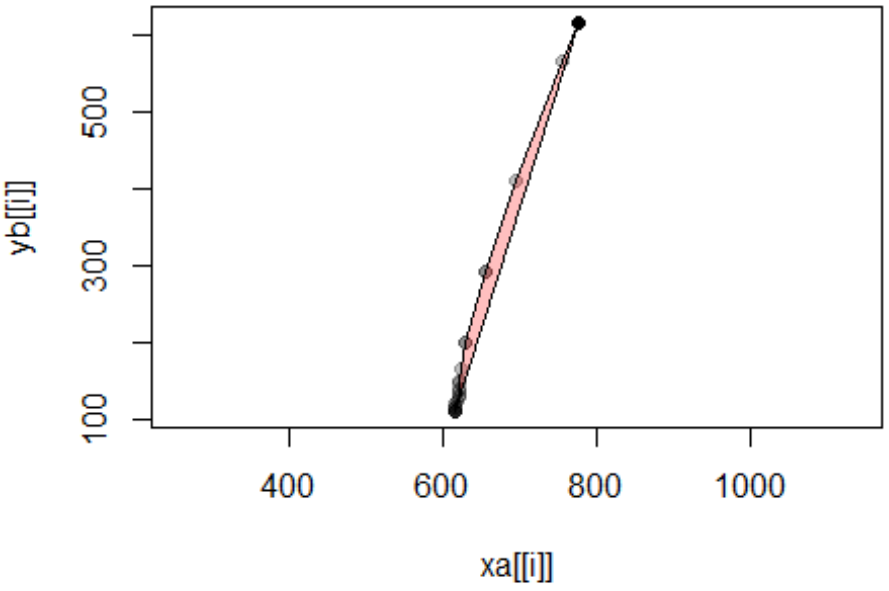

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

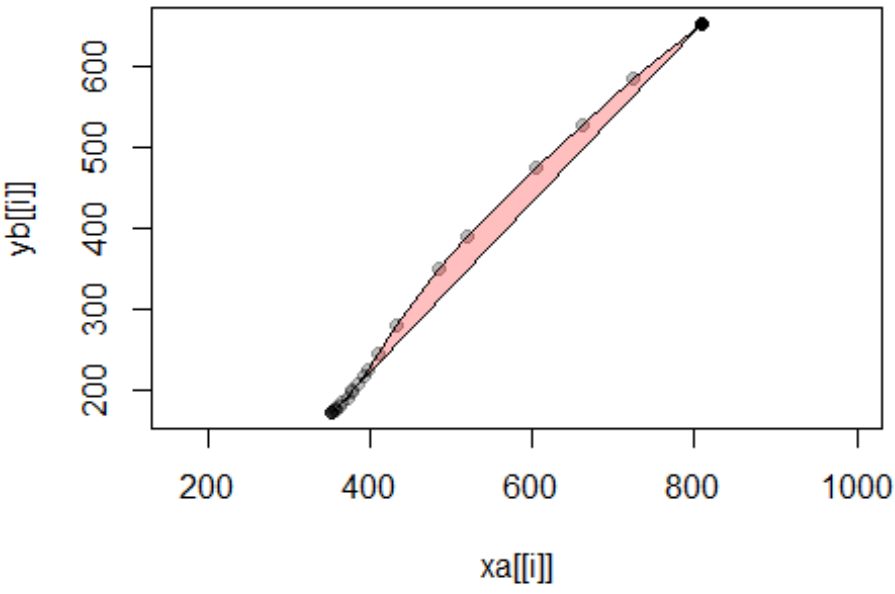

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

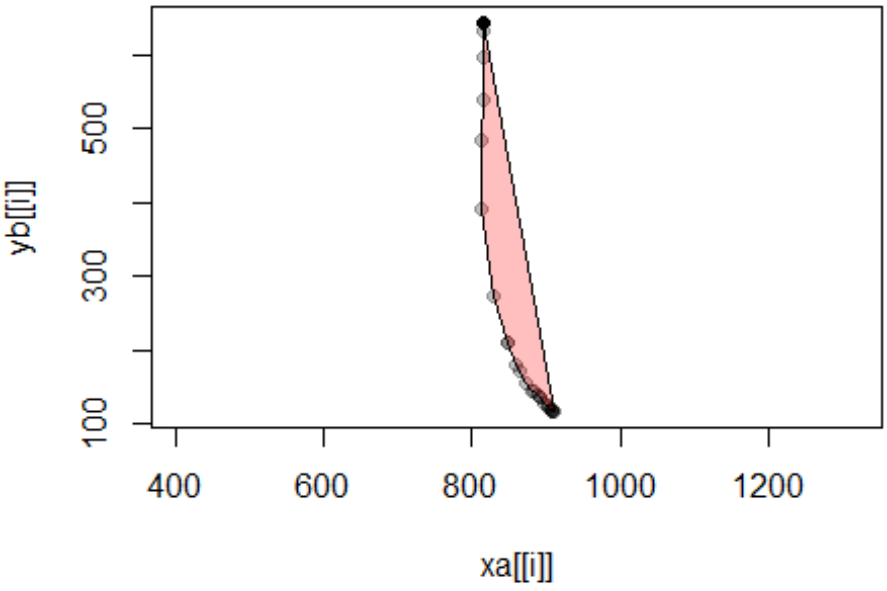

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

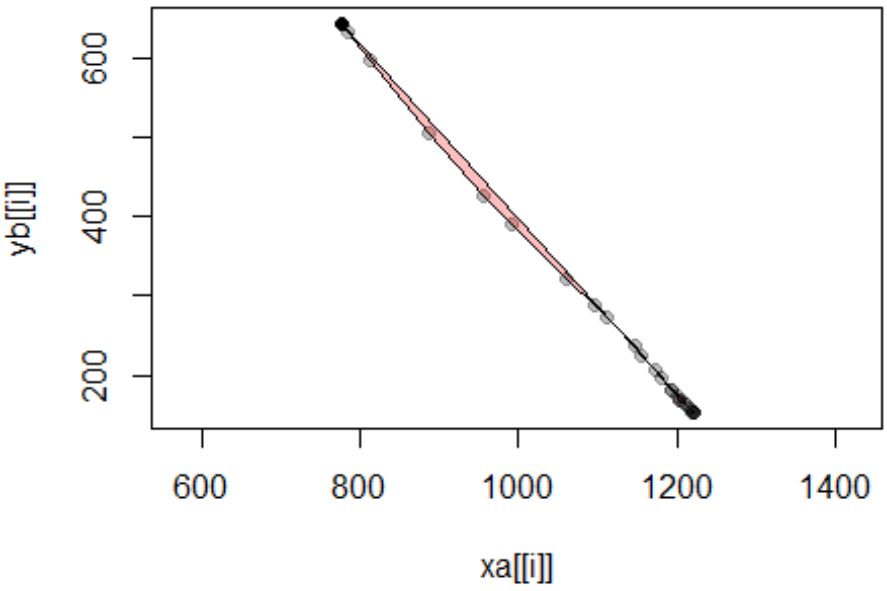

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

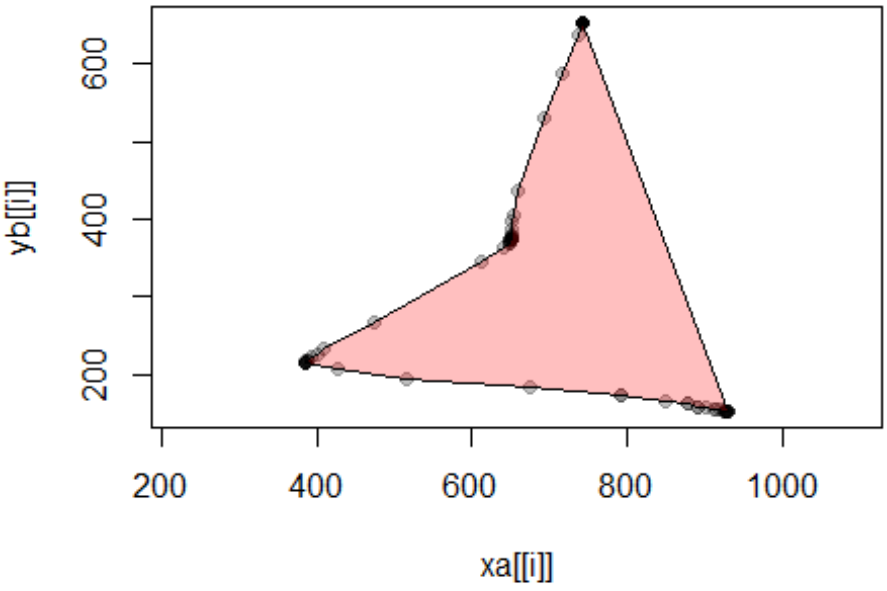

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

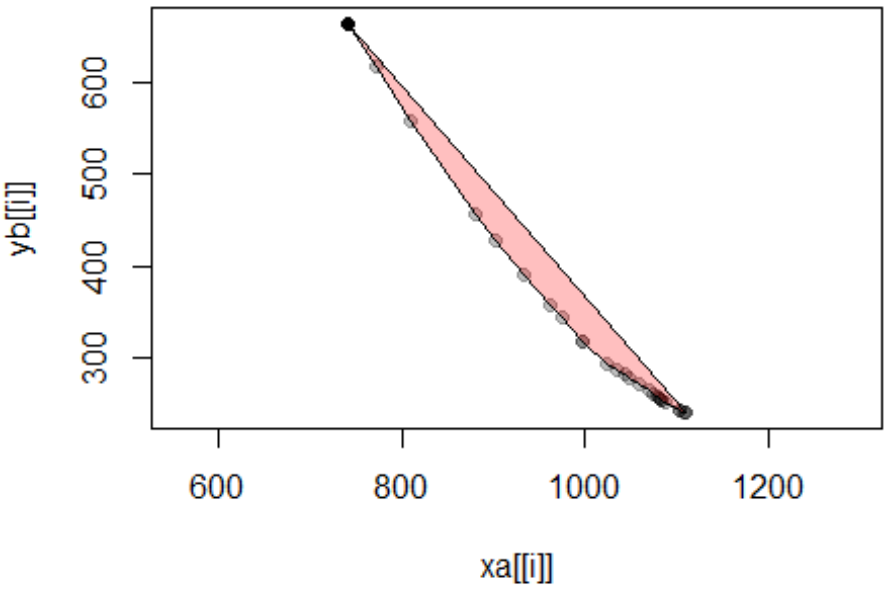

yb[[i]]
100
300
500
400
600
800
1000
xa[[i]]

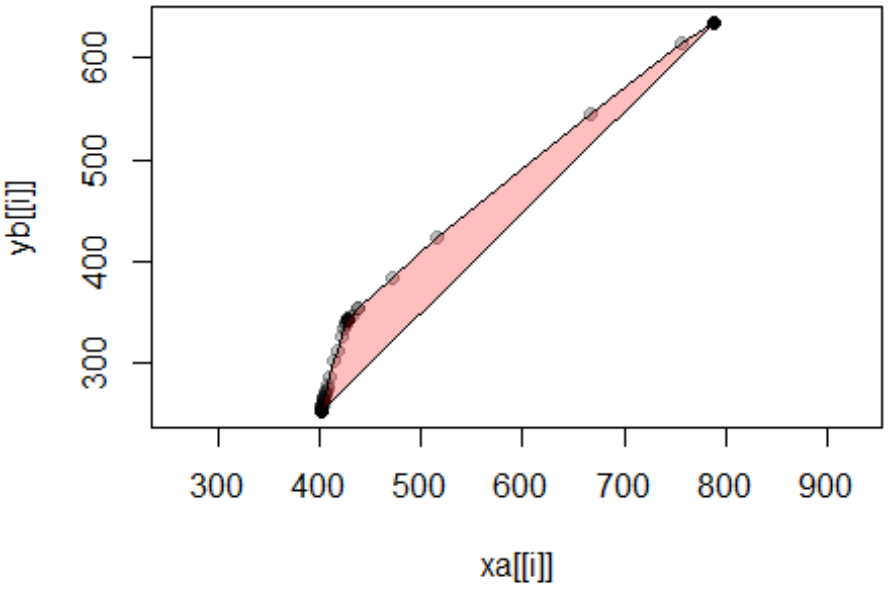

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

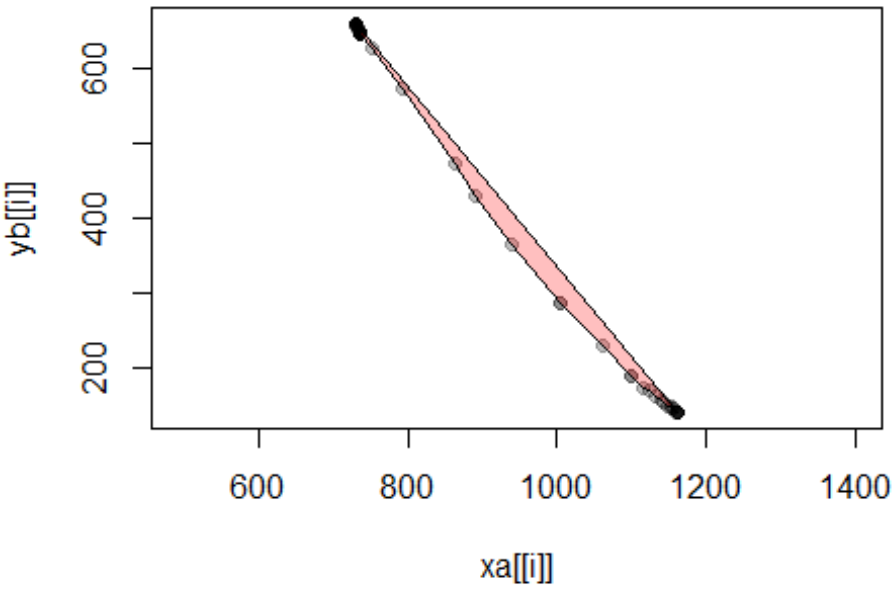

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

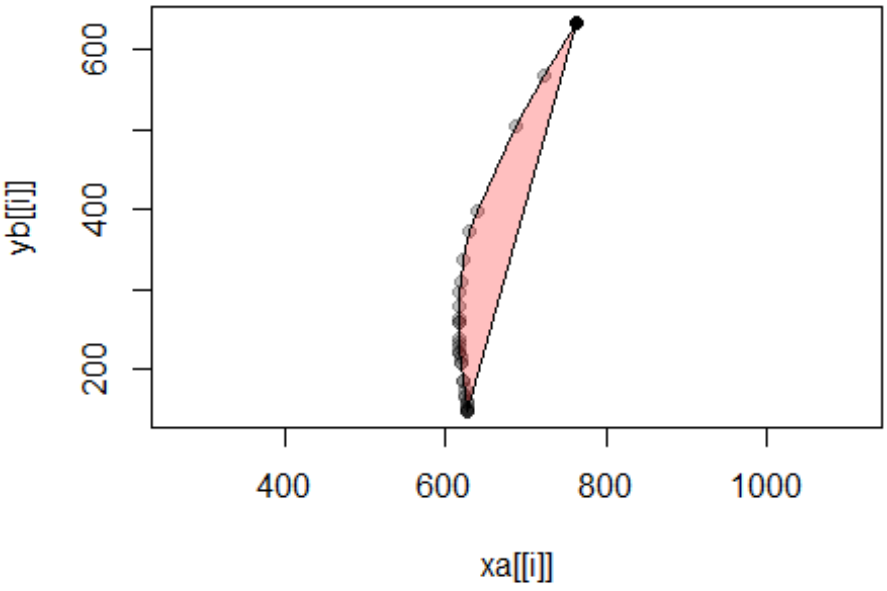

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

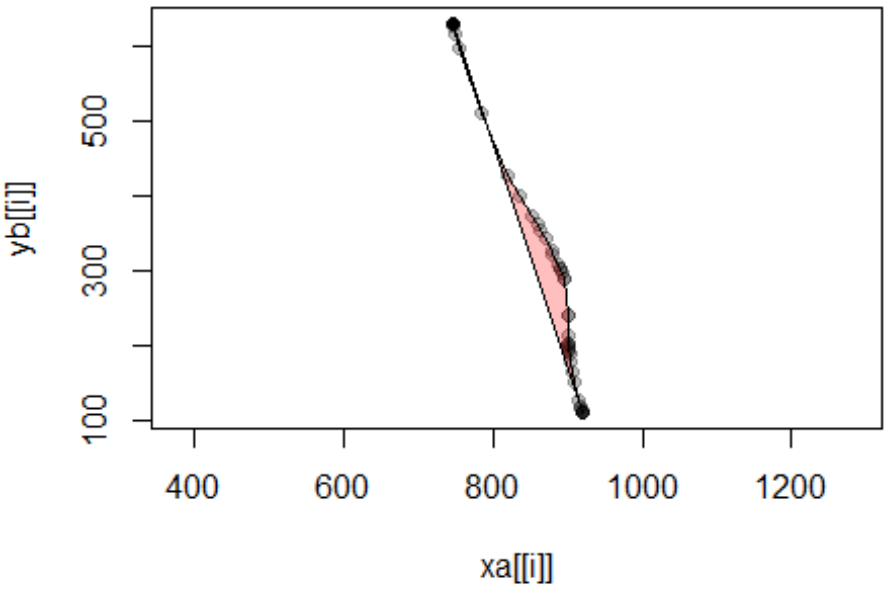

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

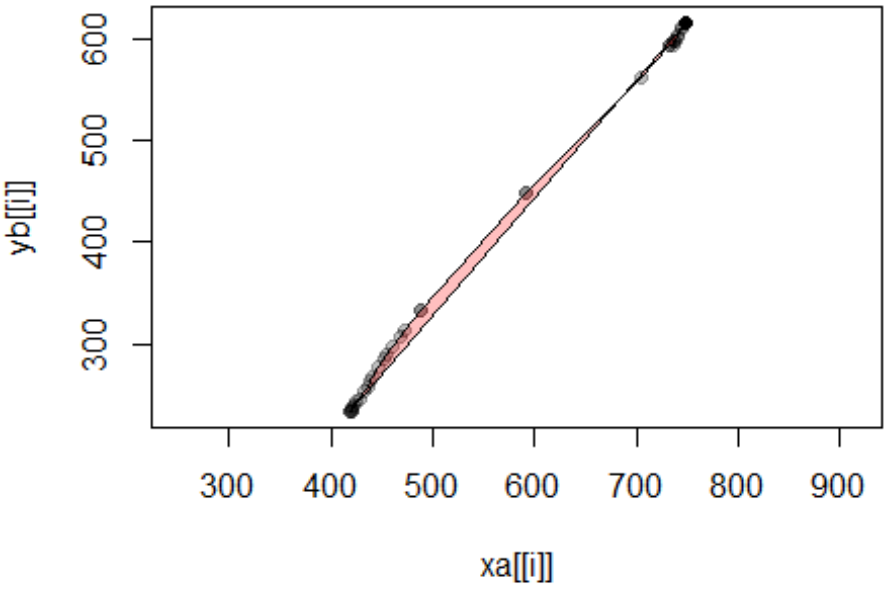

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

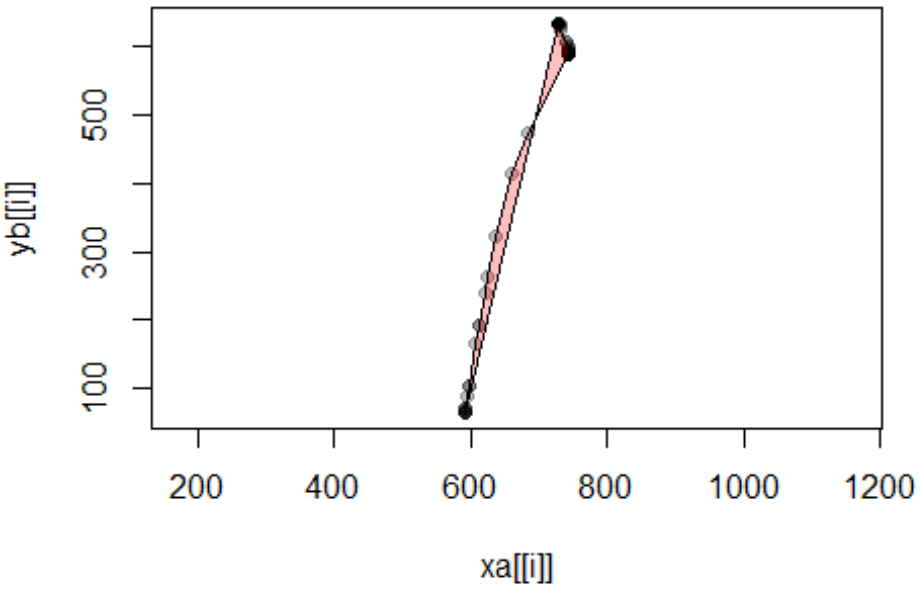

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

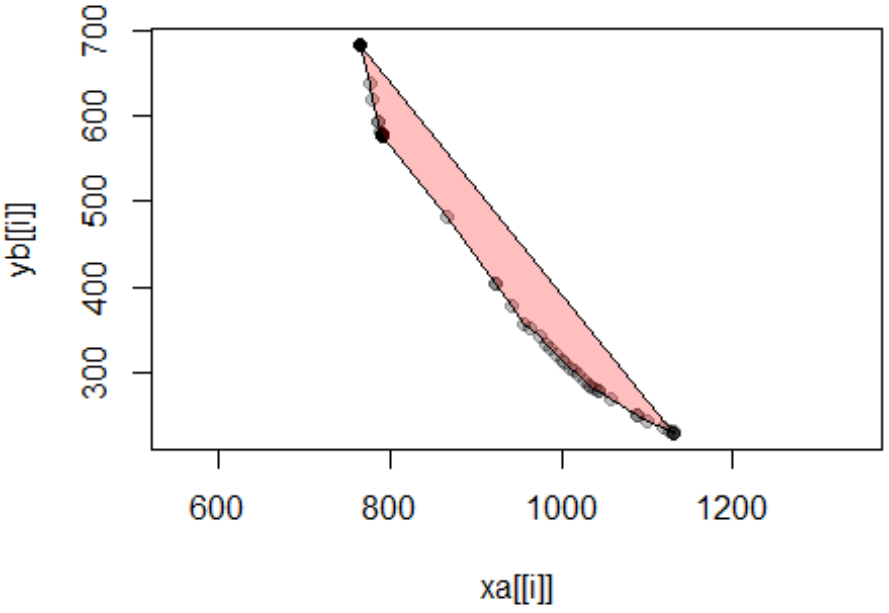

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

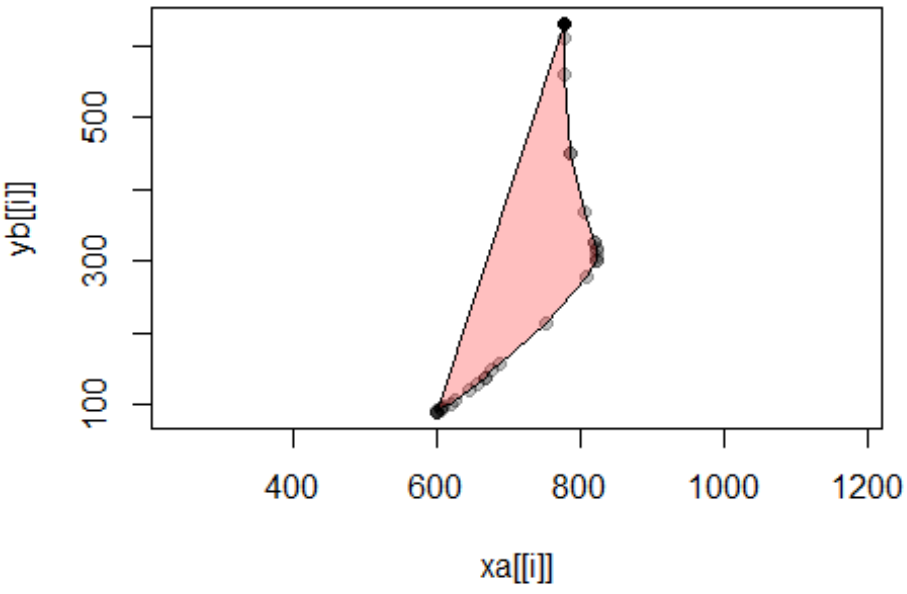

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

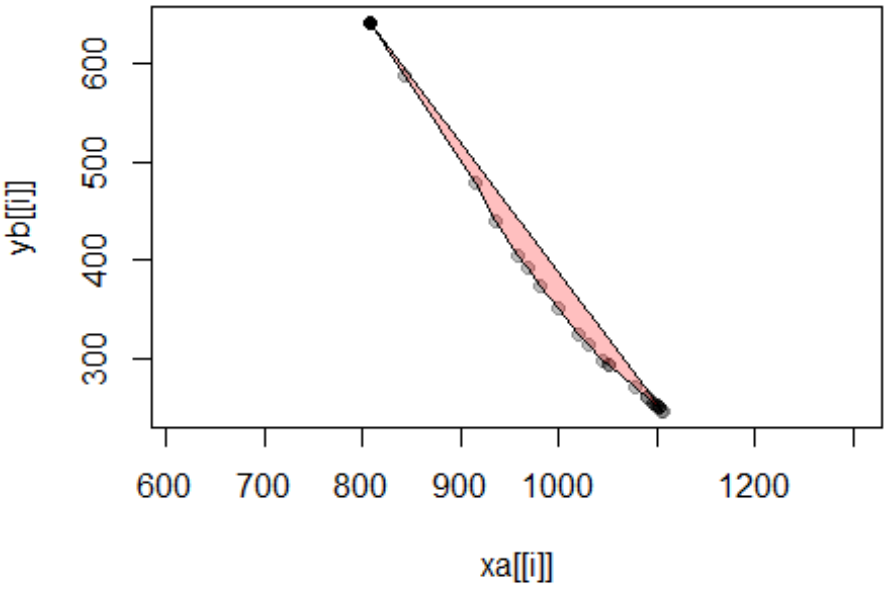

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

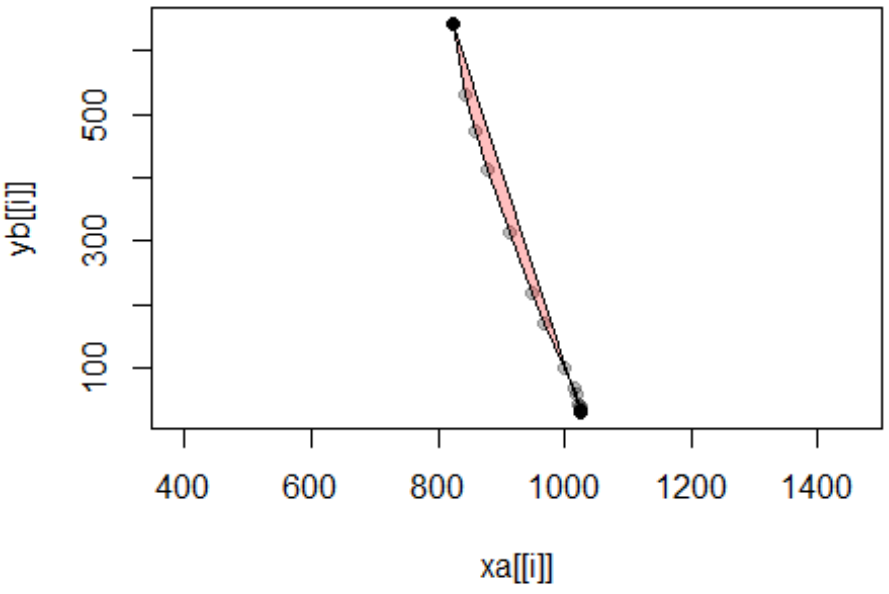

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

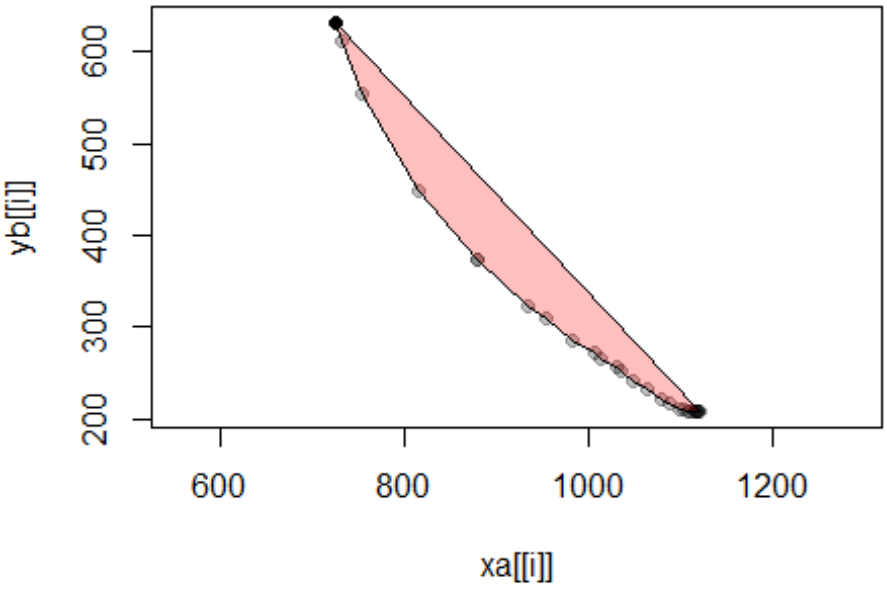

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

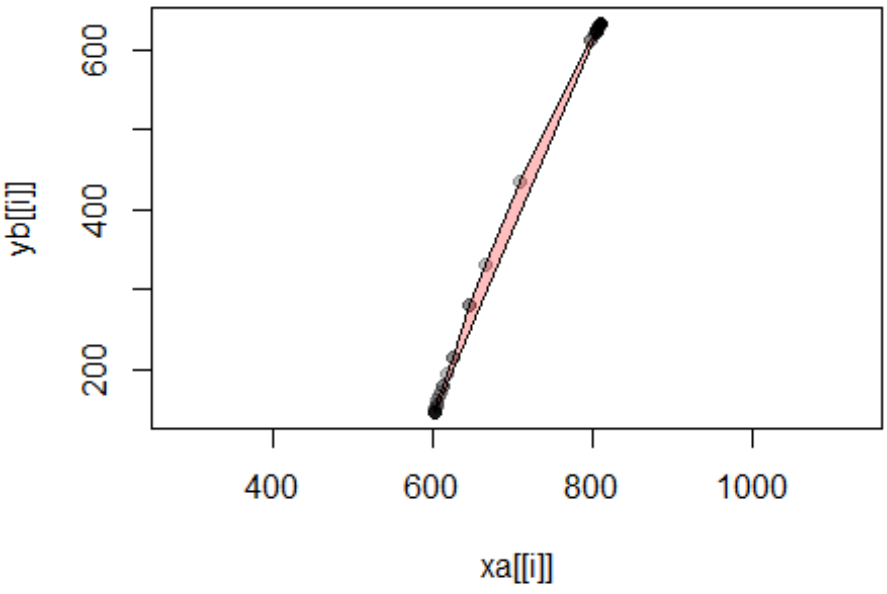

yb[[i]]
700
600
500
400
300
600
800
1000
1200
xa[[i]]

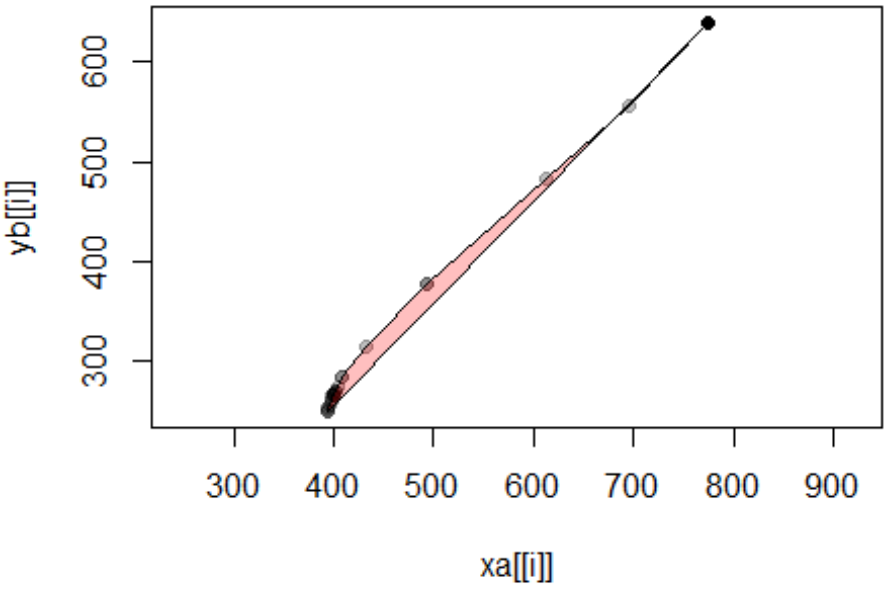

yb[[i]]
500
300
100
400
600
800
1000
1200
xa[[i]]

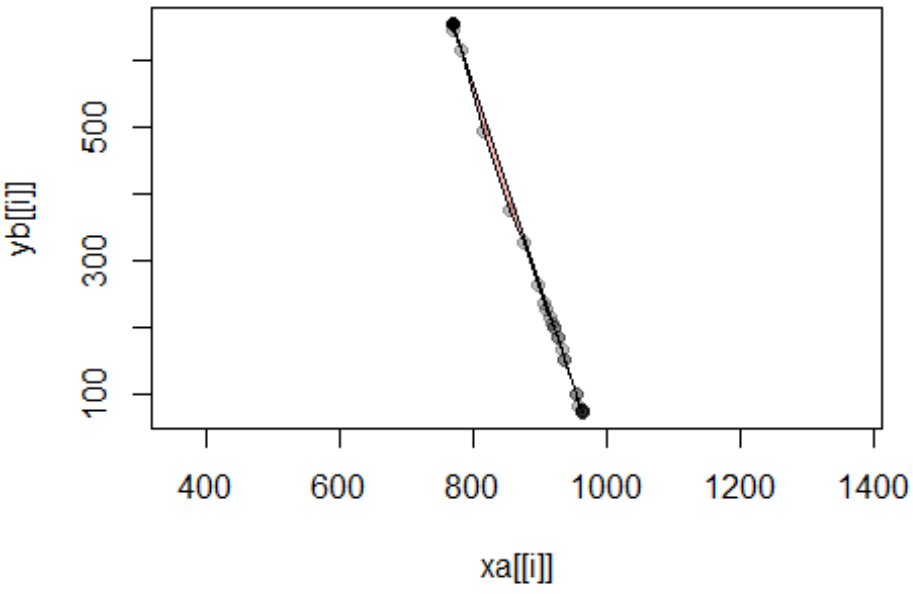

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

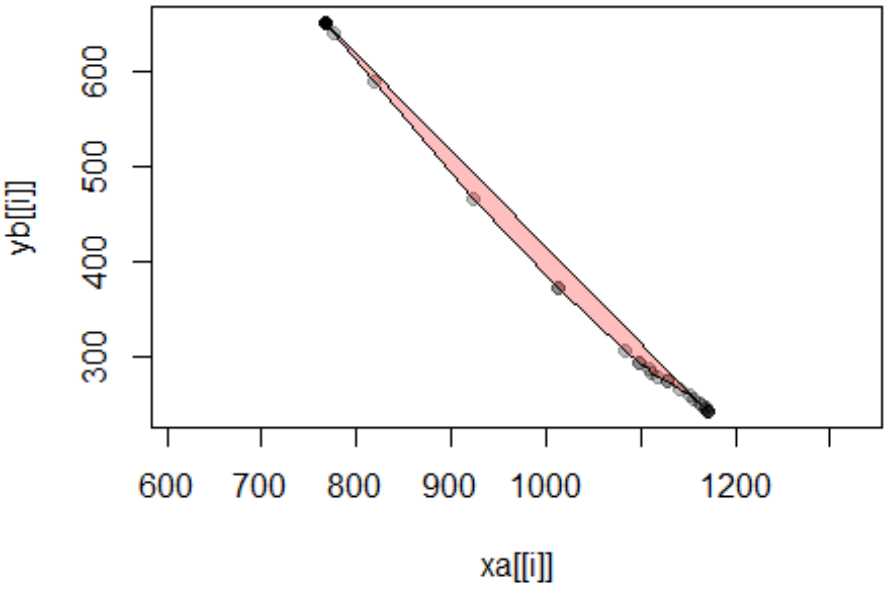

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

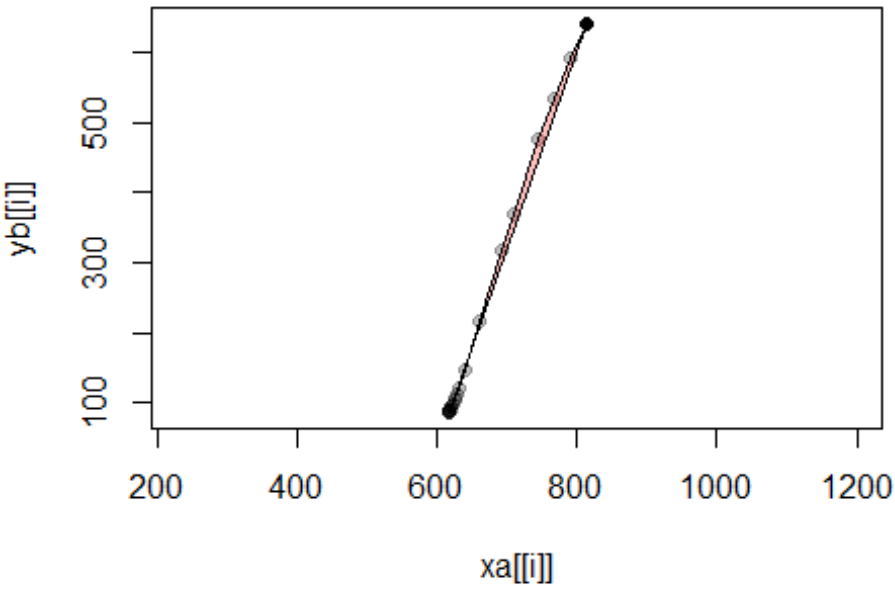

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

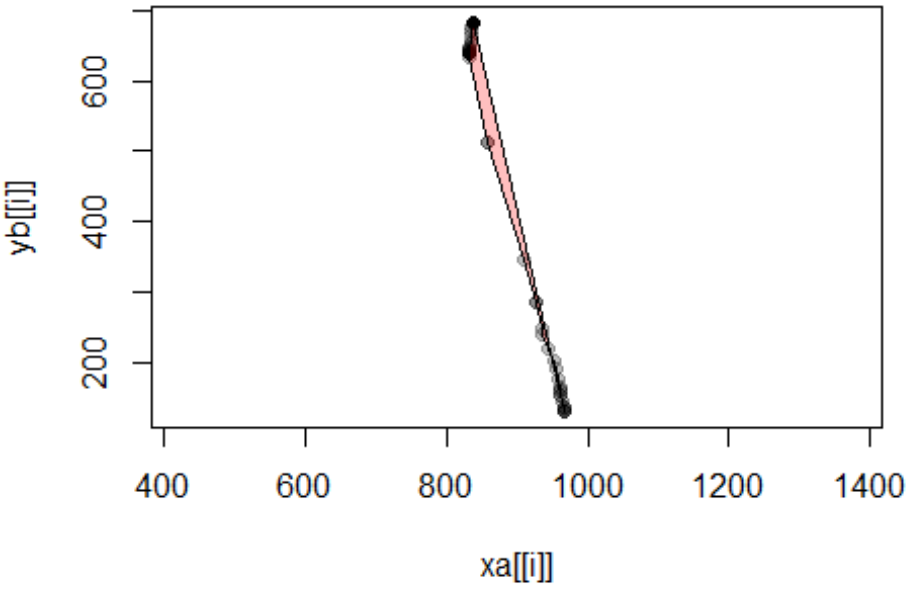

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

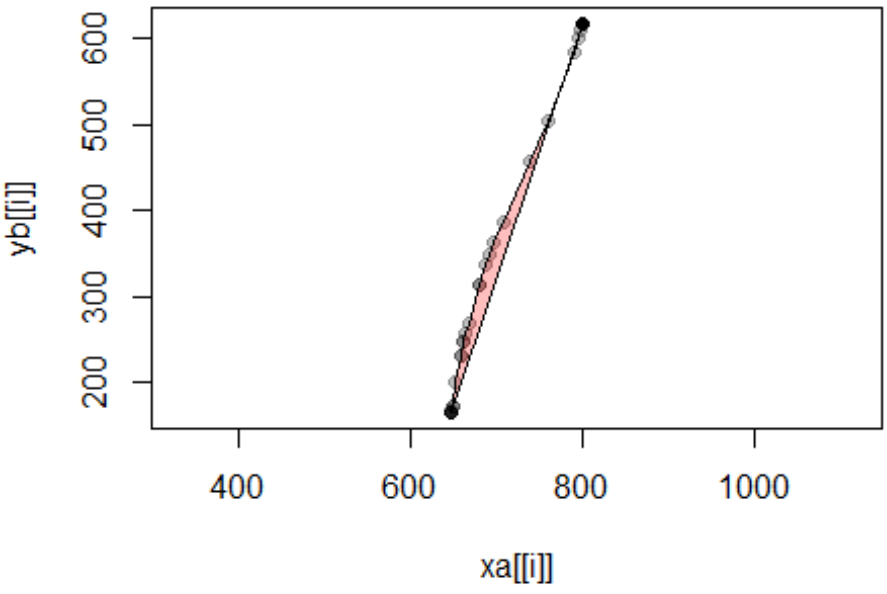

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

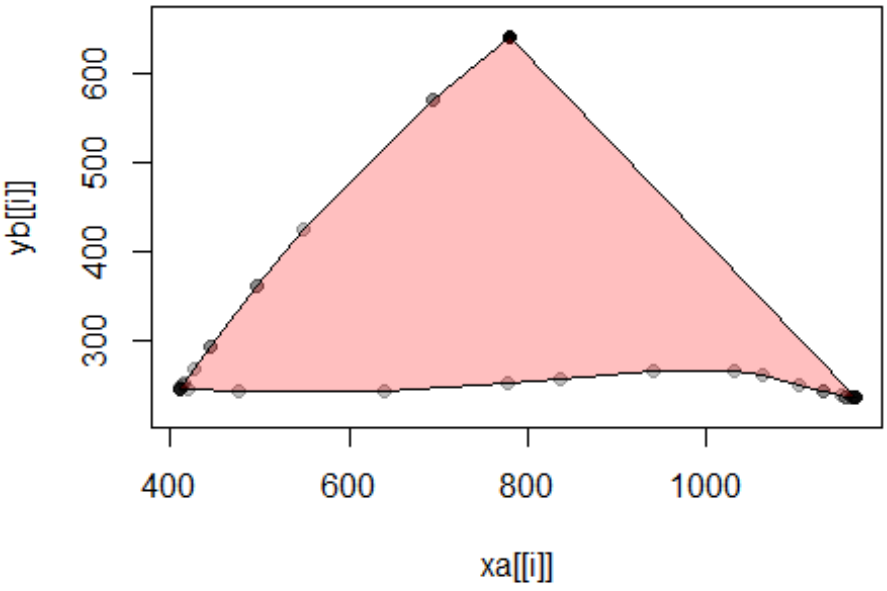

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

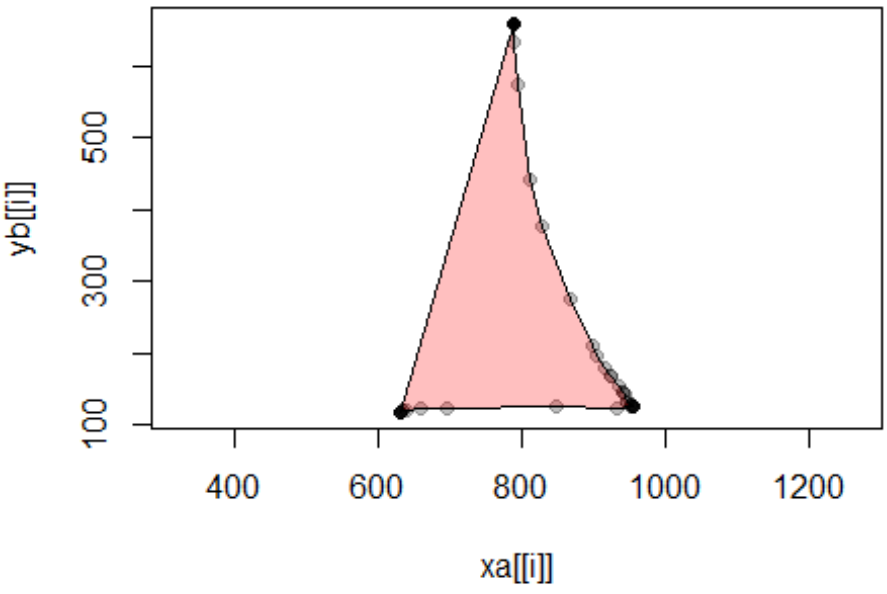

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

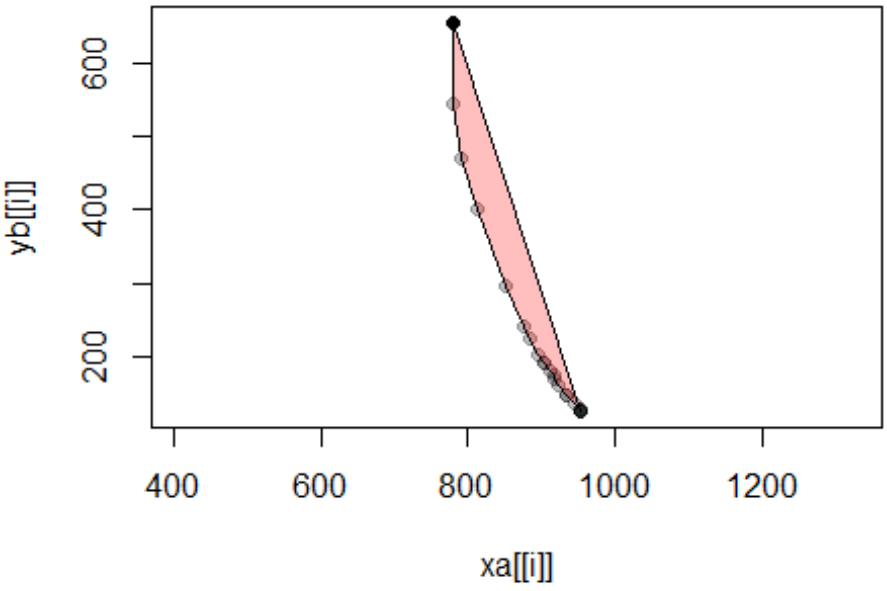

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

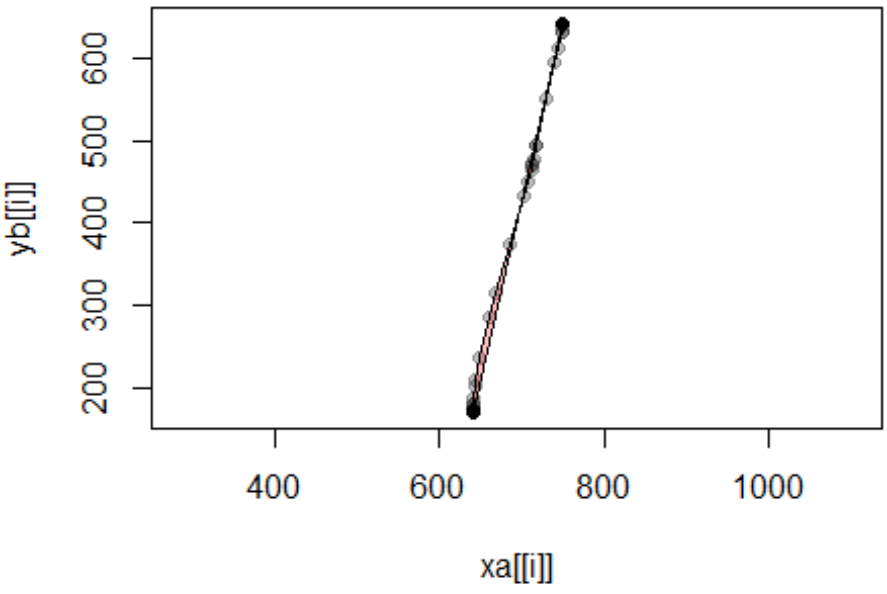

yb[[i]]
200
400
600
400
600
800
1000
1200
1400
xa[[i]]

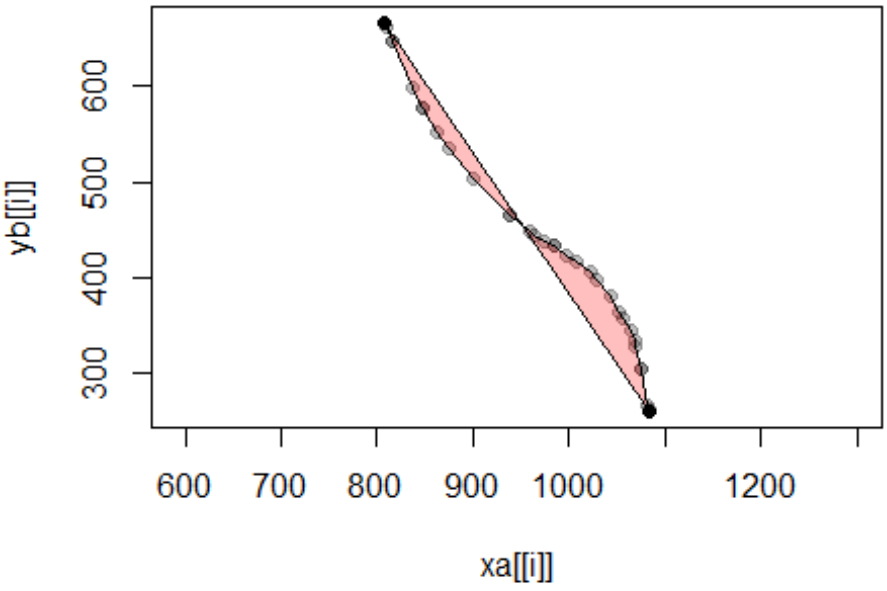

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

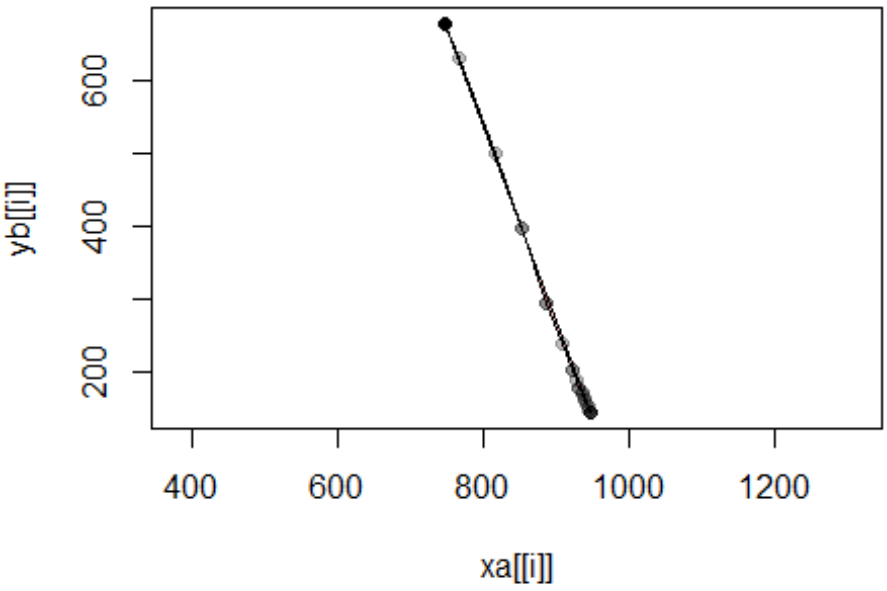

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

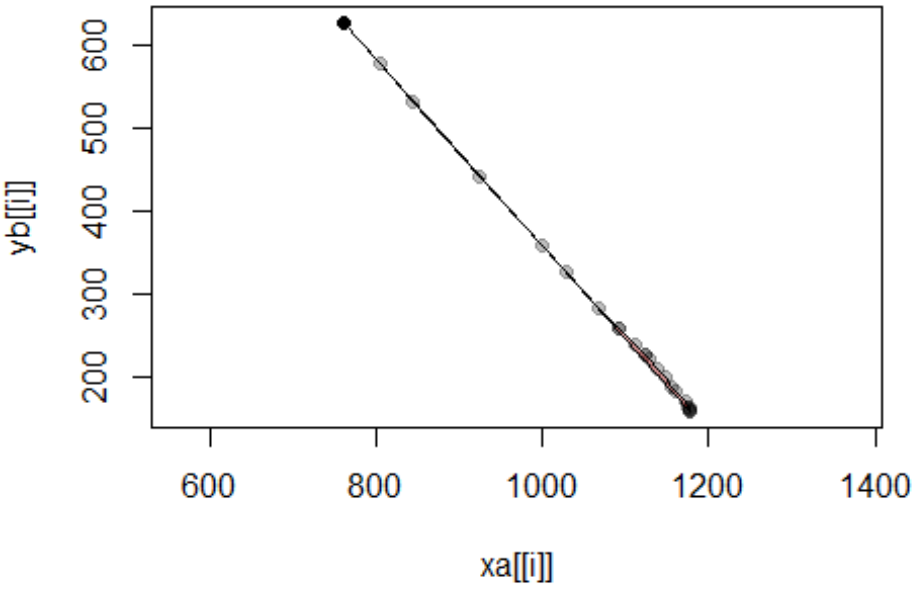

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

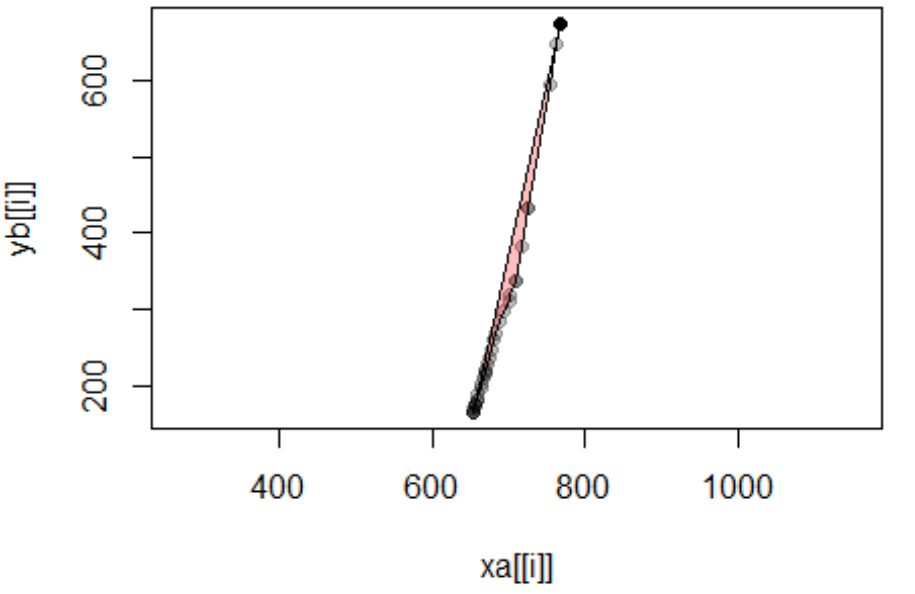

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

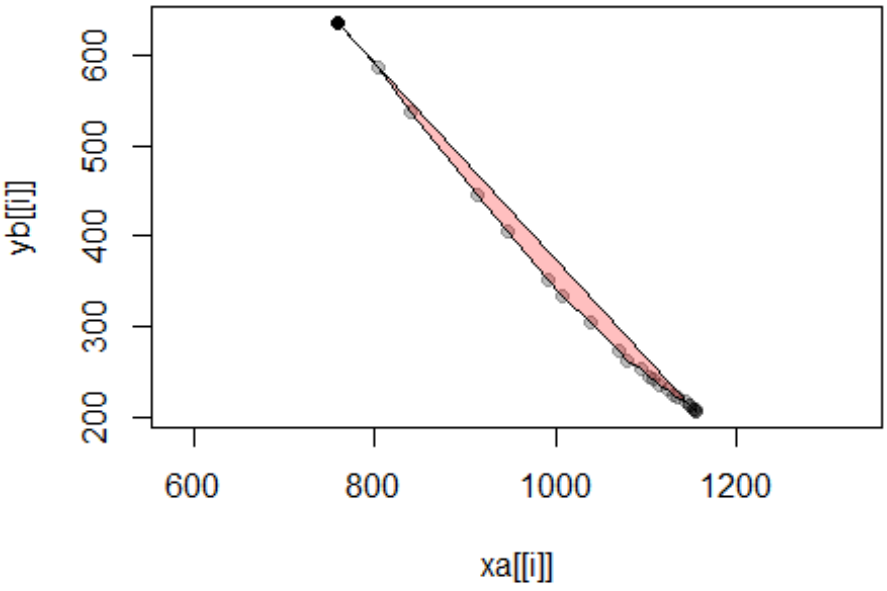

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

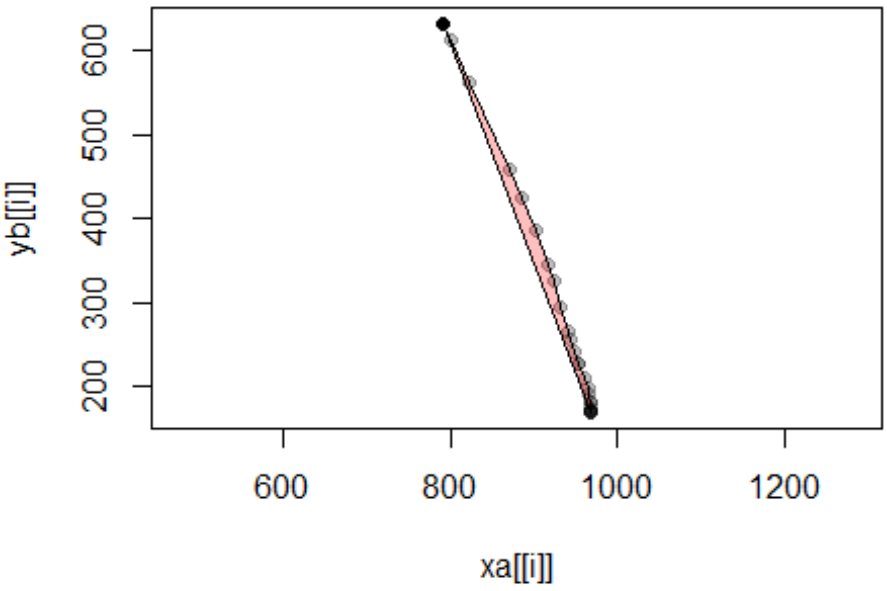

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

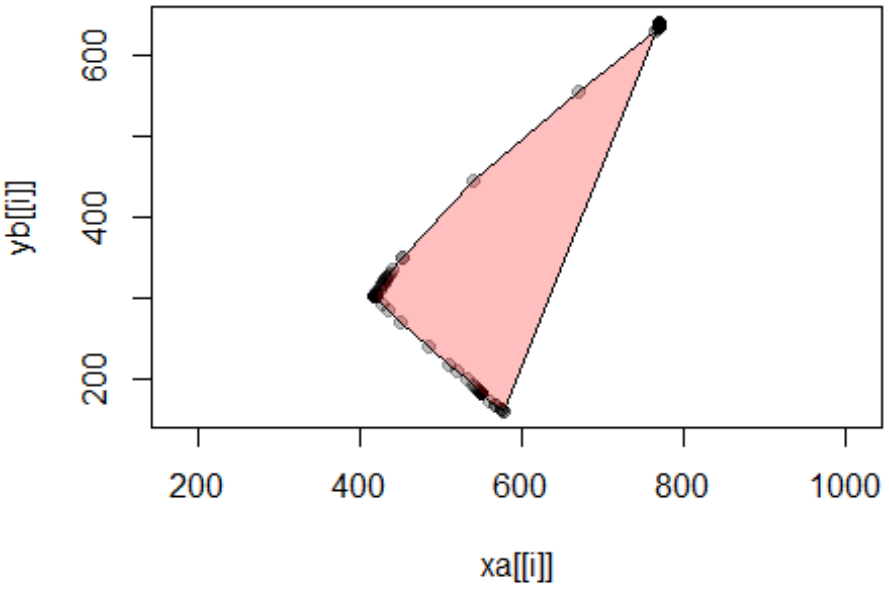

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

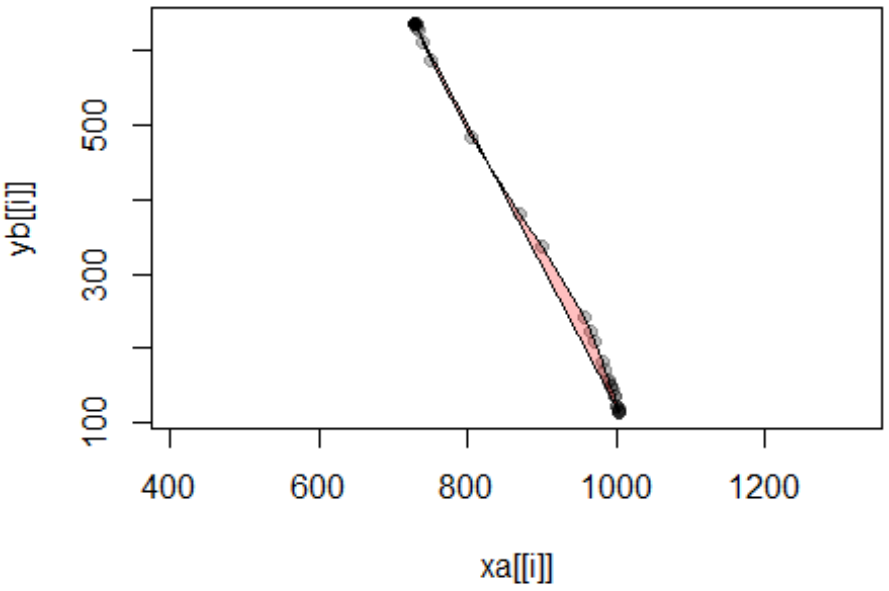

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

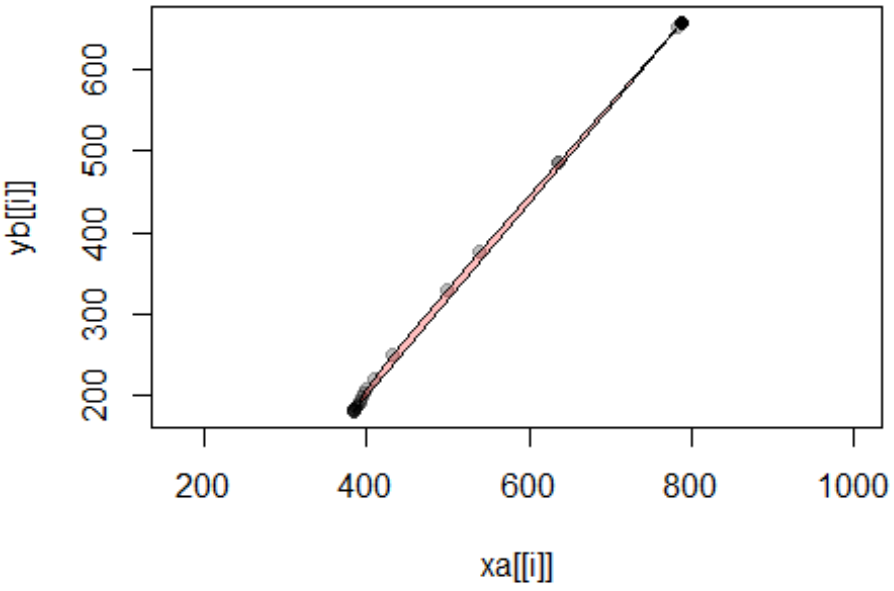

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

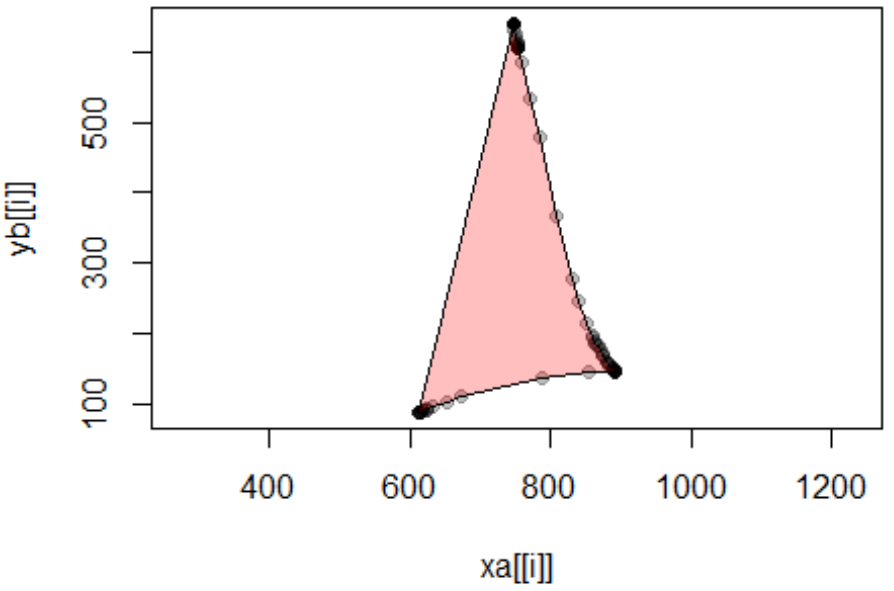

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

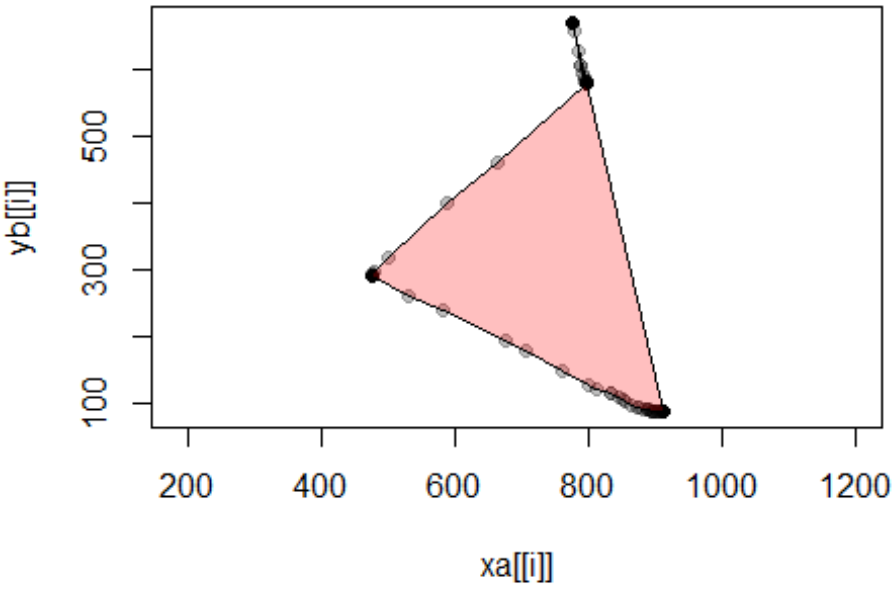

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

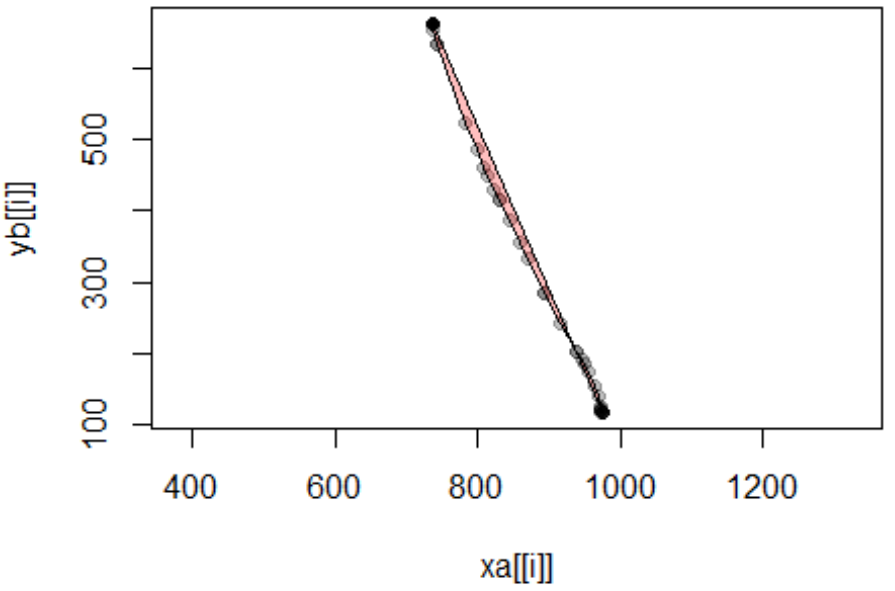

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

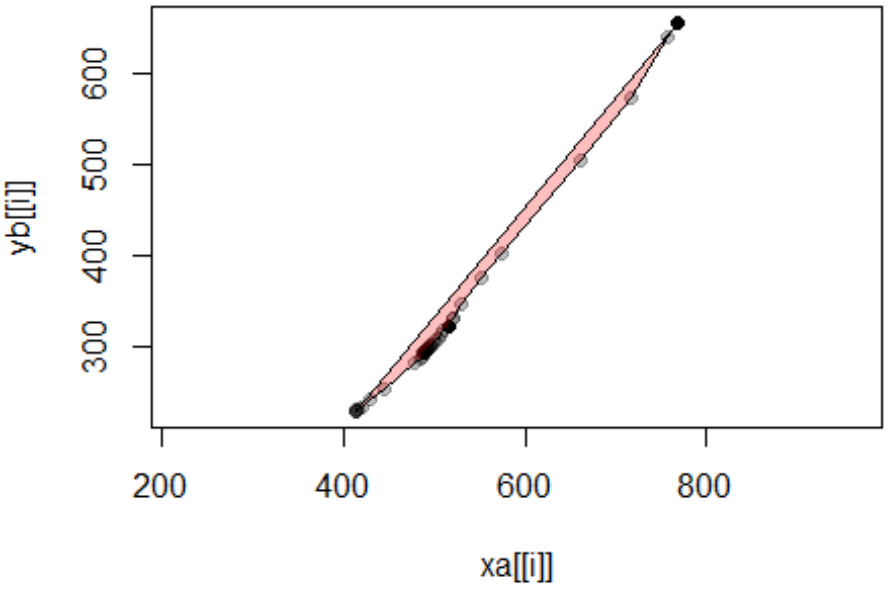

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

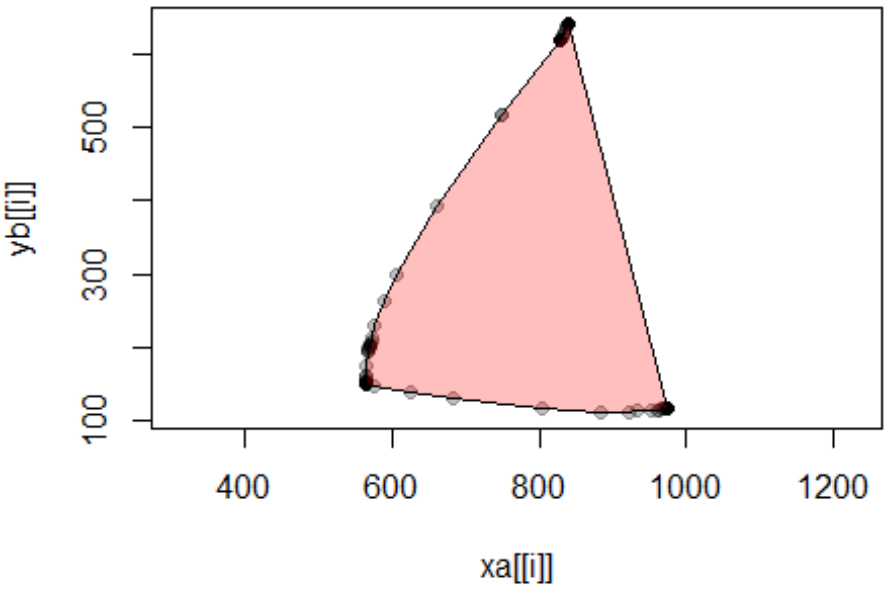

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

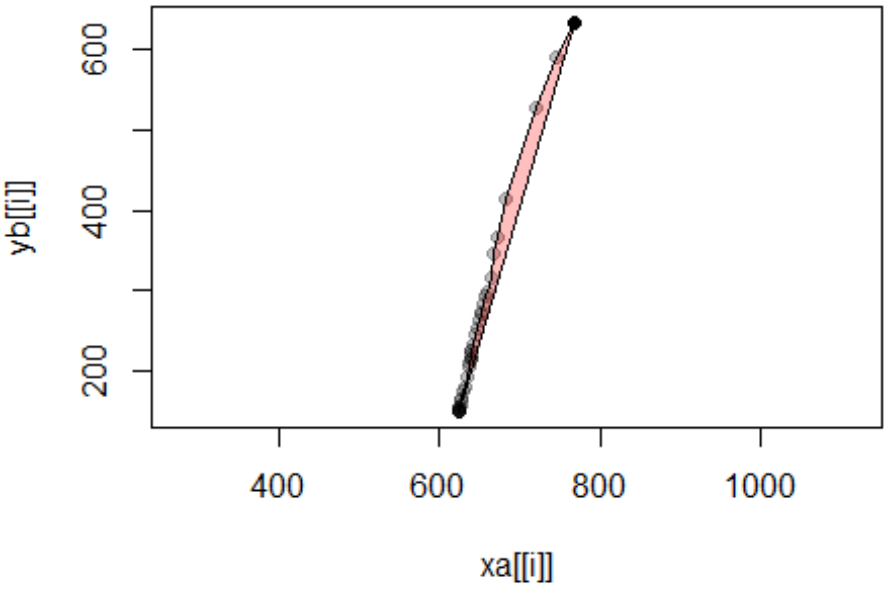

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

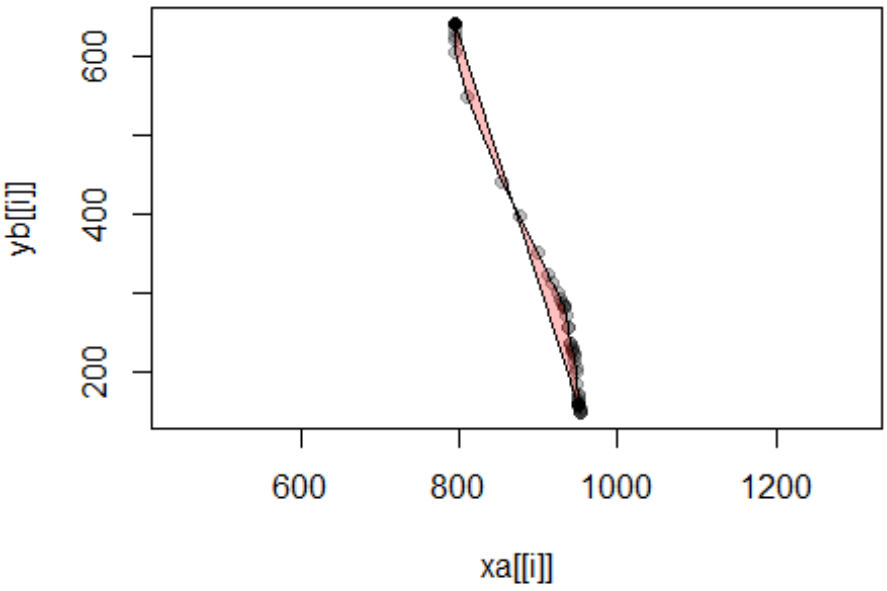

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

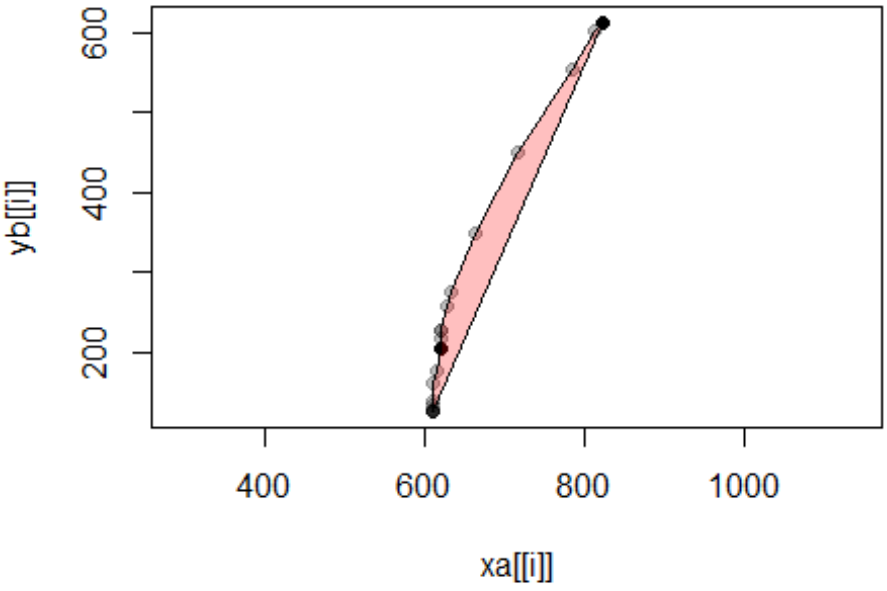

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

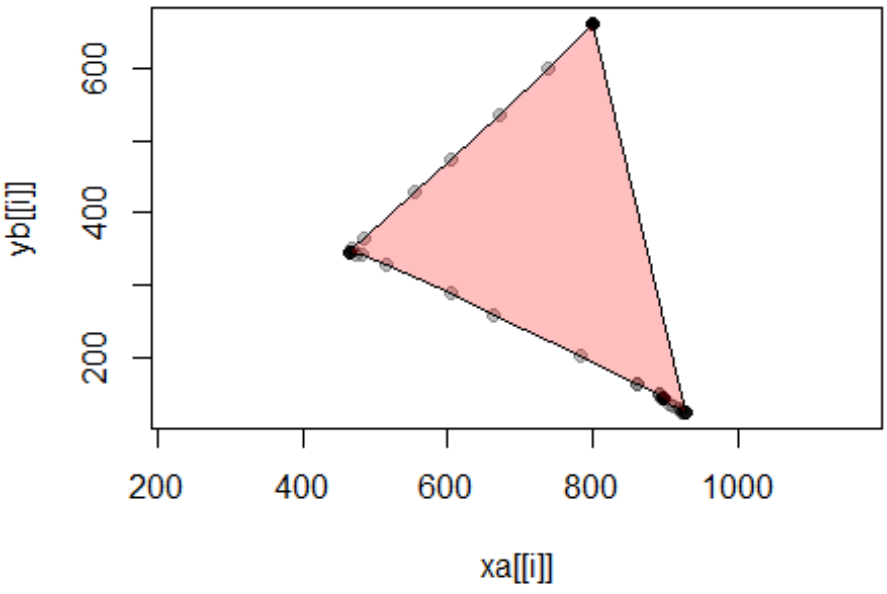

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

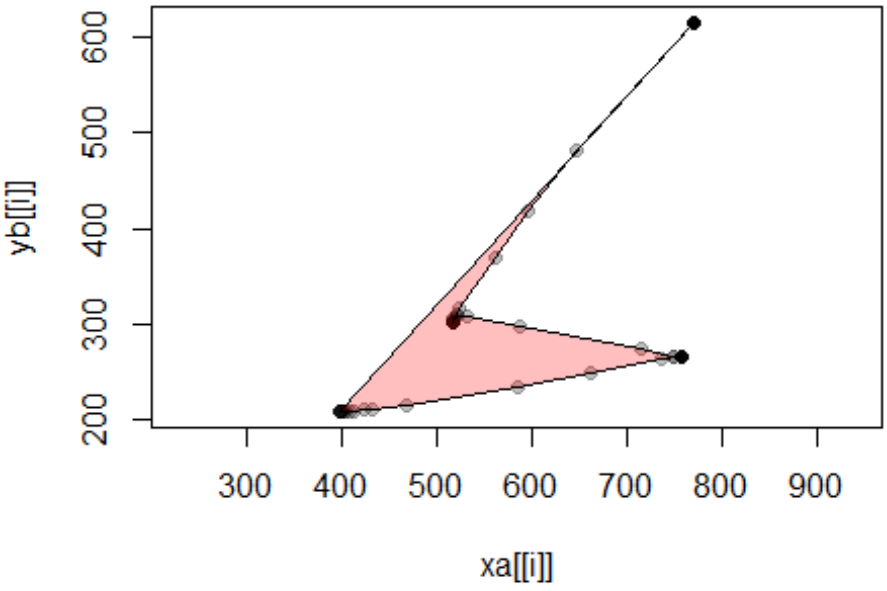

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

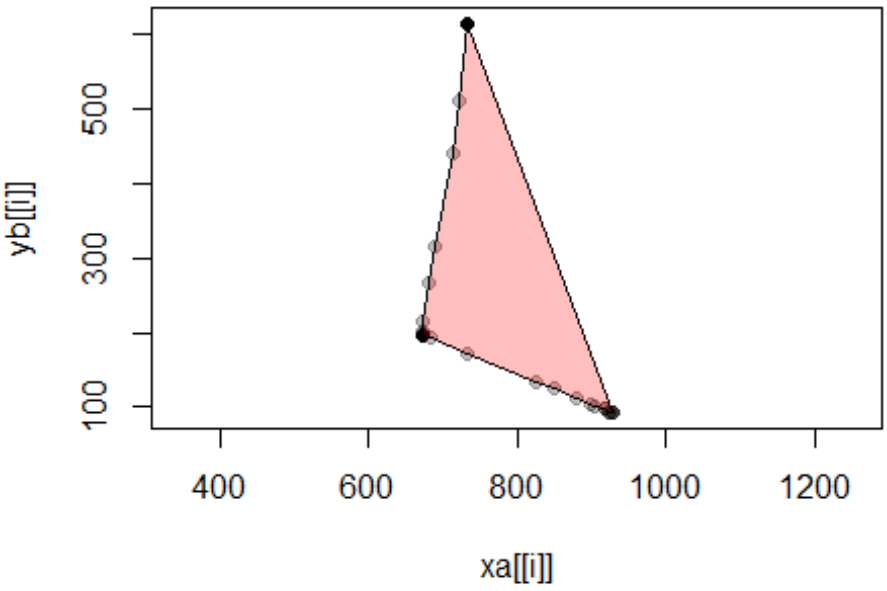

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

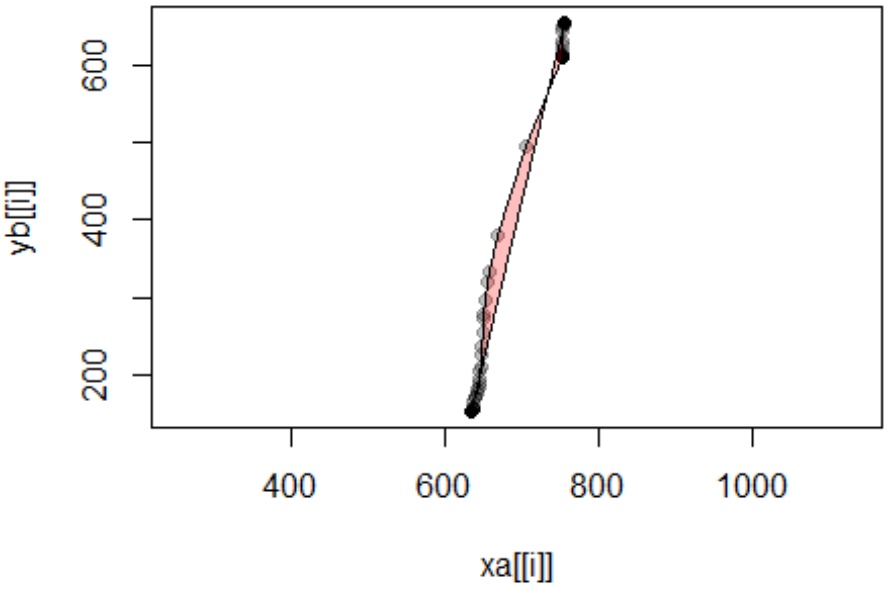

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

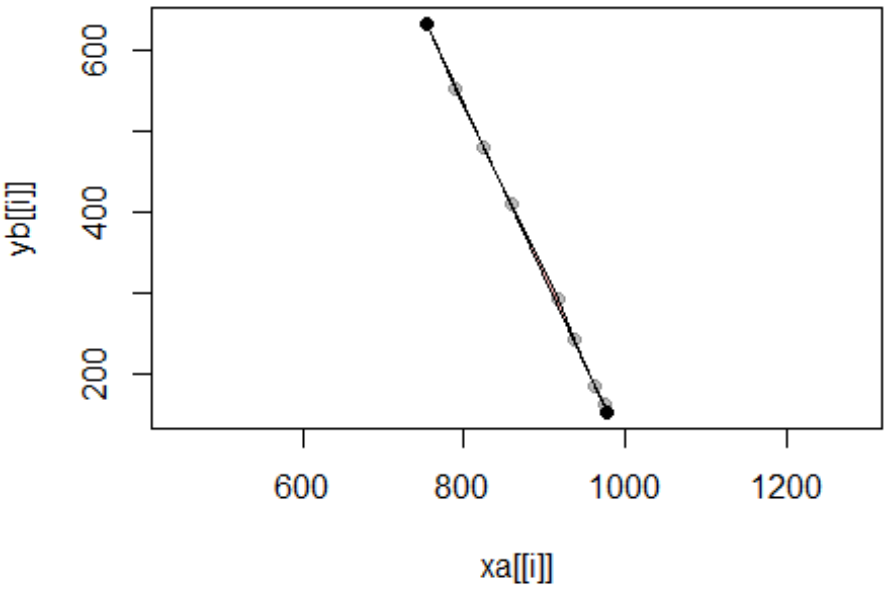

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

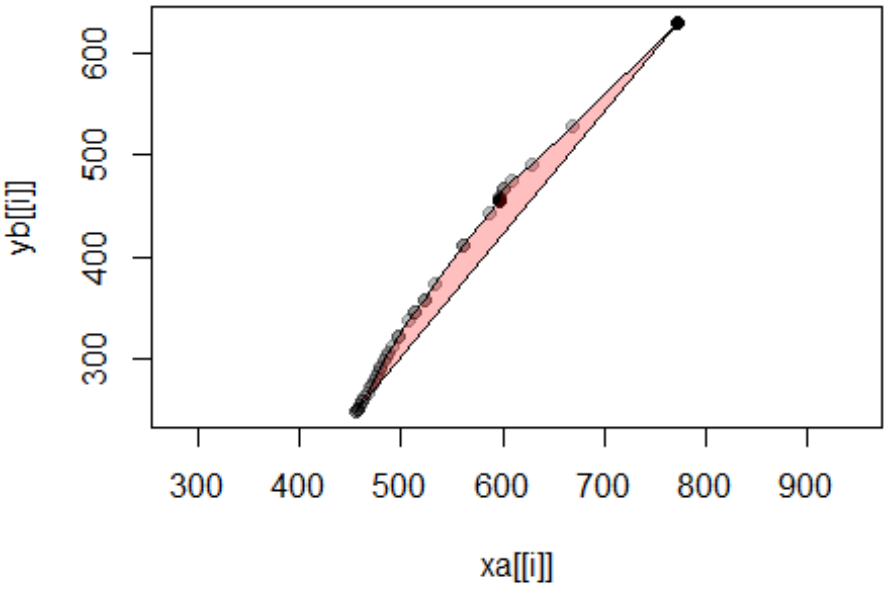

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

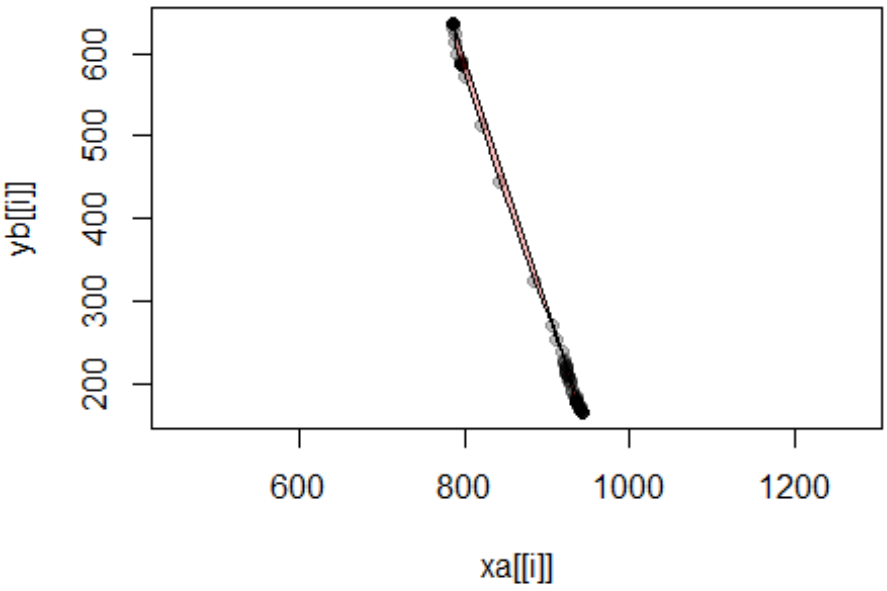

yb[[i]]
200
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

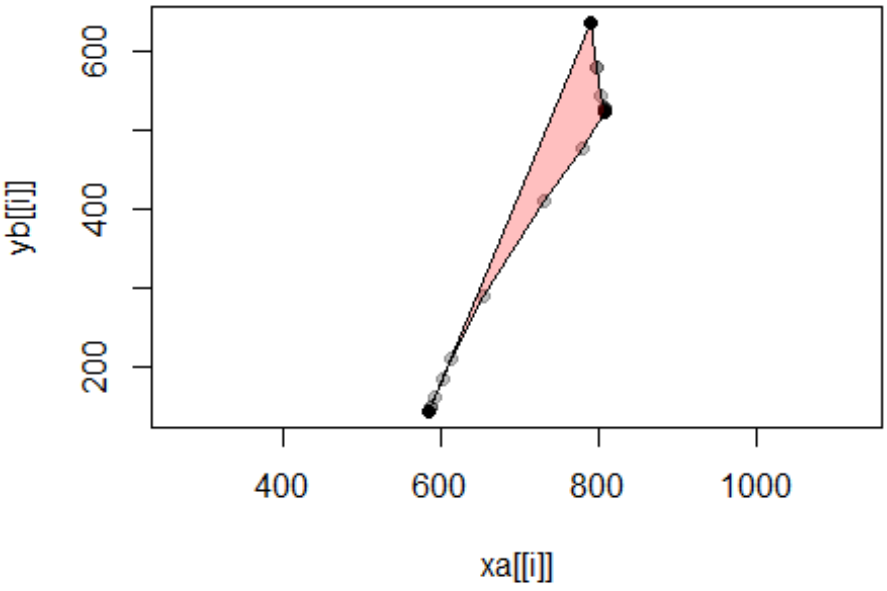

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

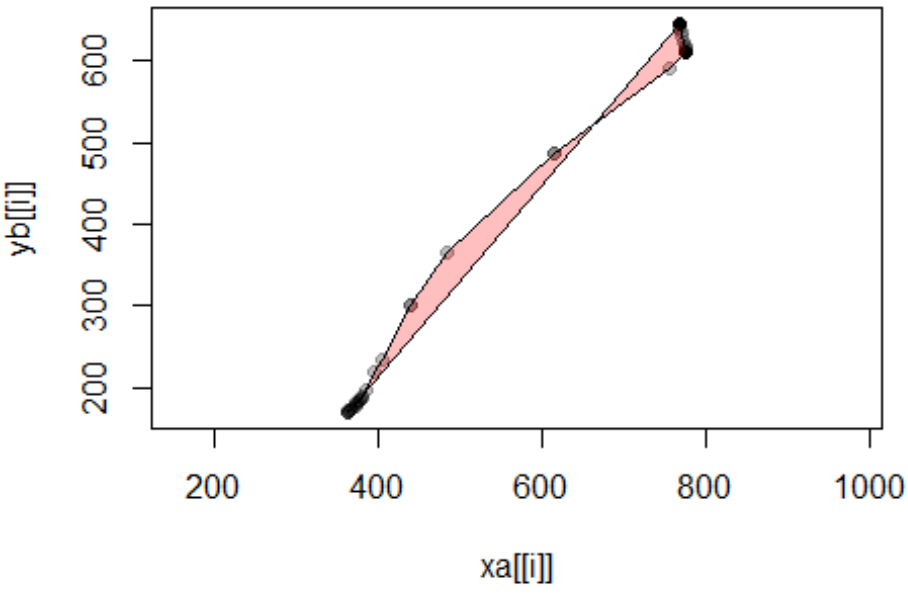

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

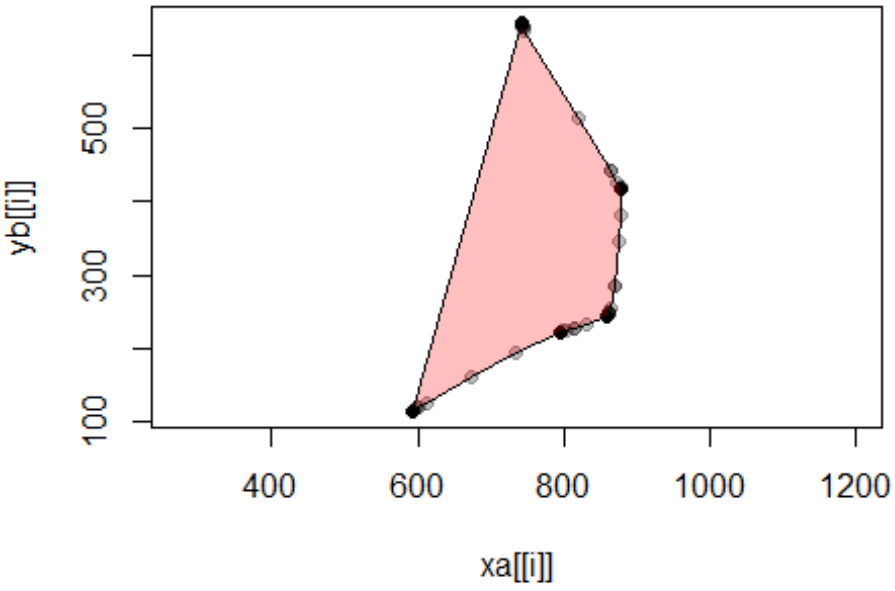

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

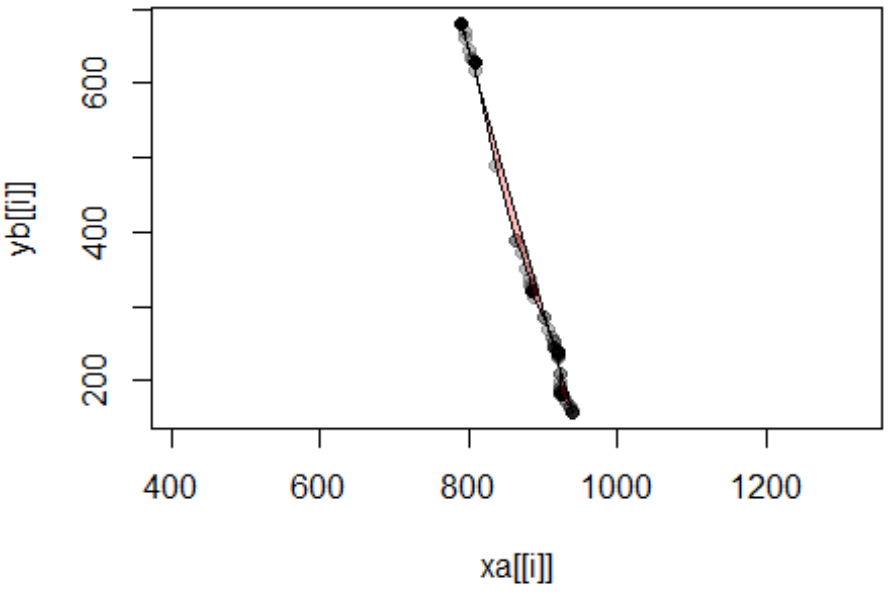

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

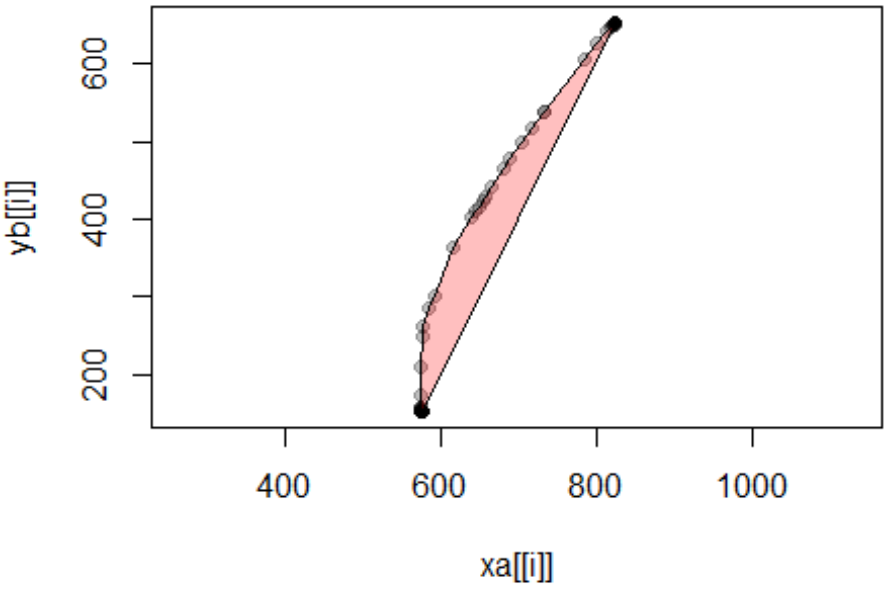

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

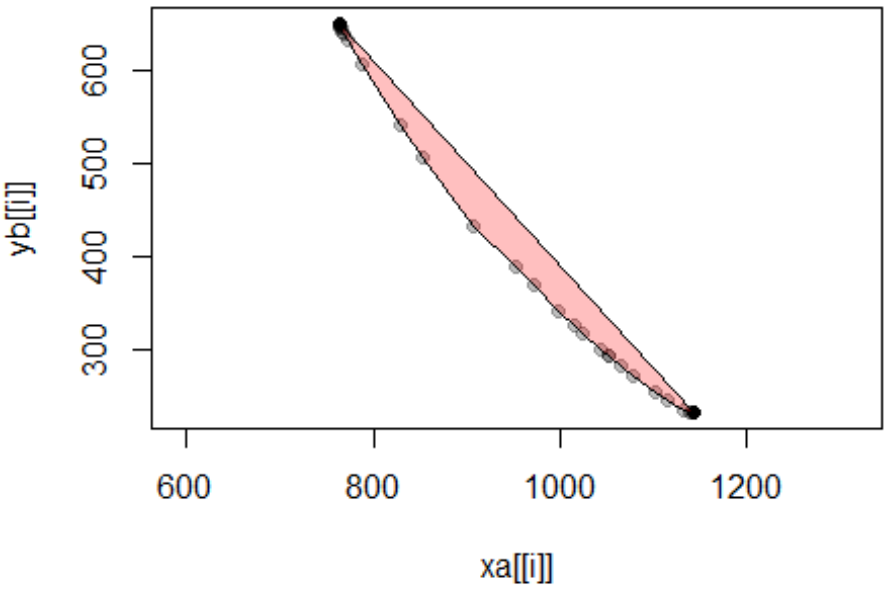

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

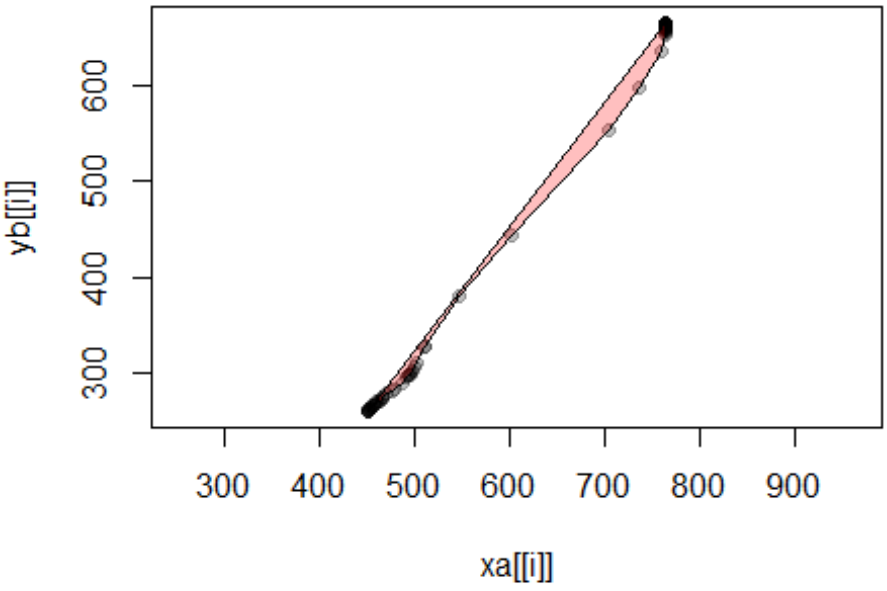

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

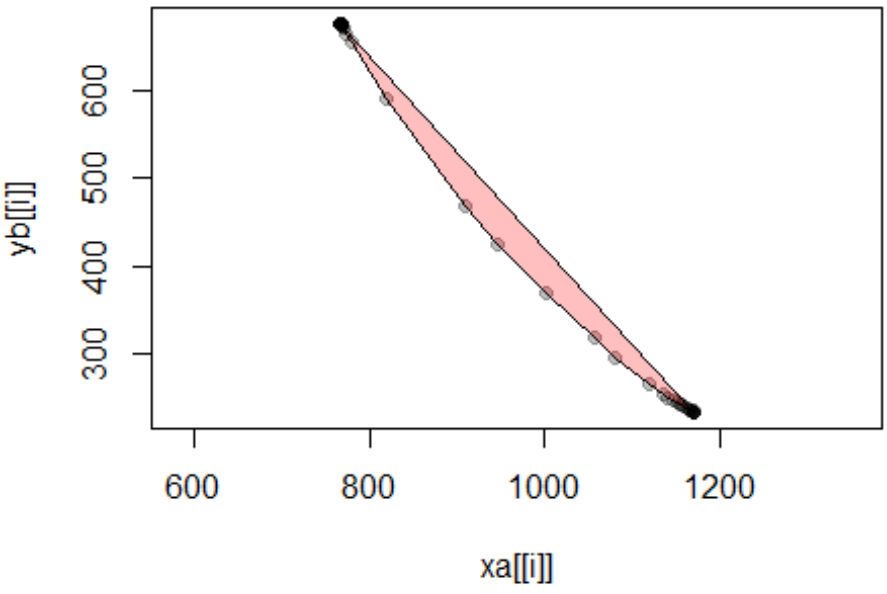

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

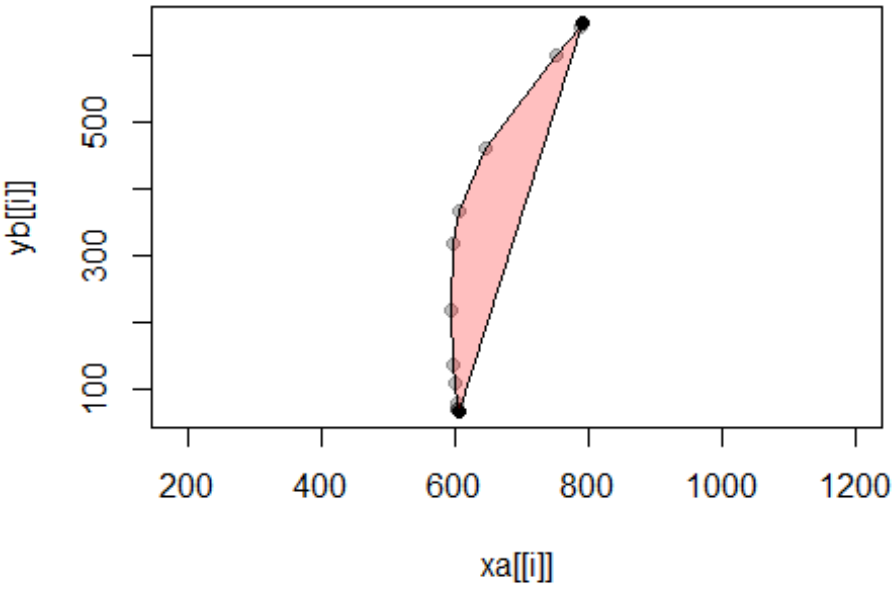

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

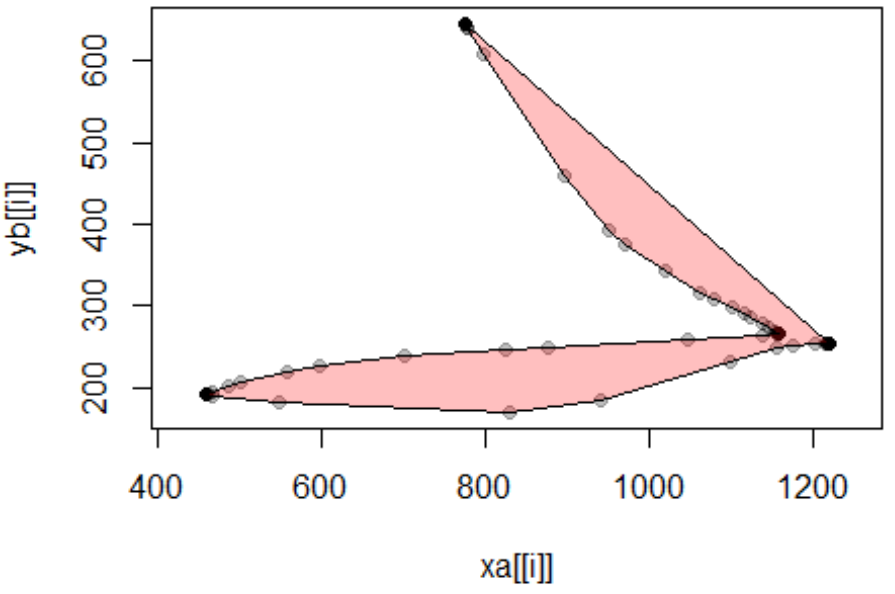

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

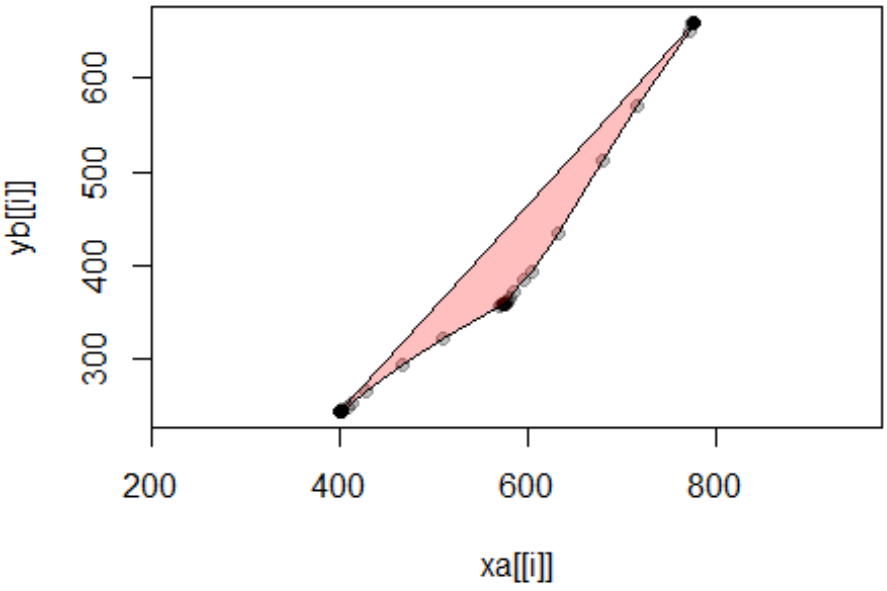

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

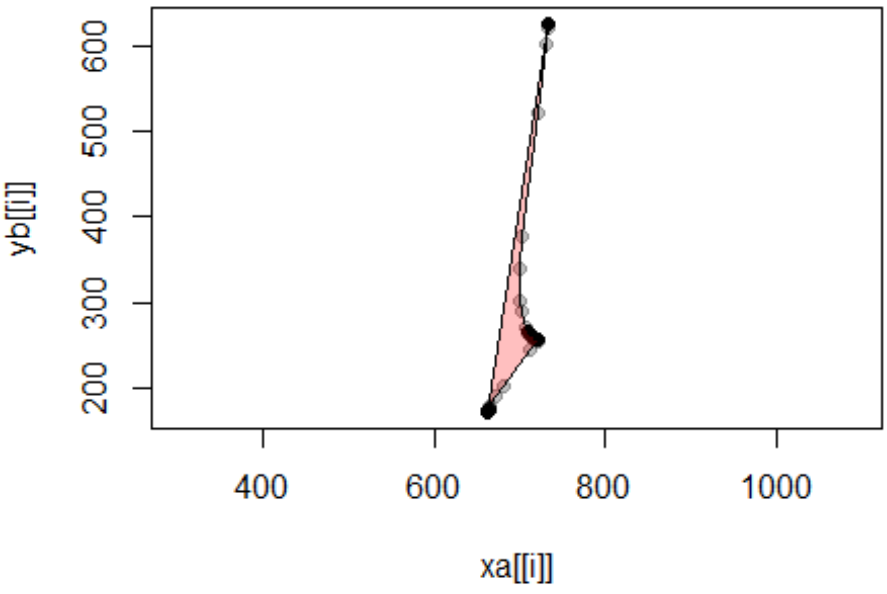

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

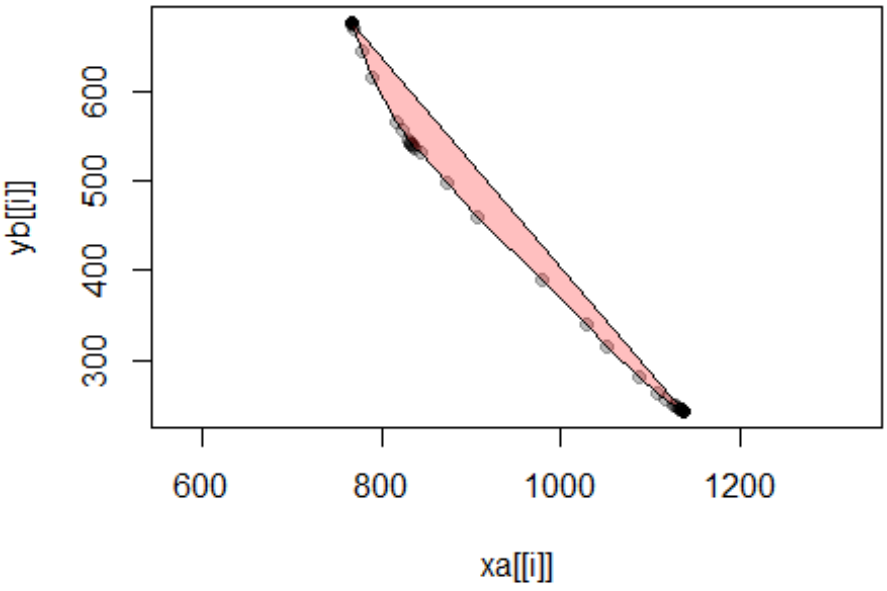

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

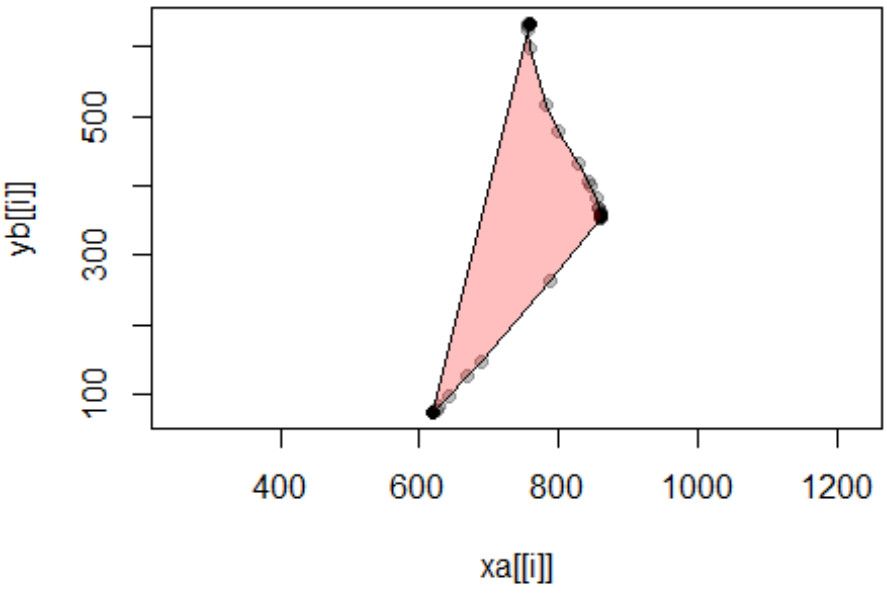

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

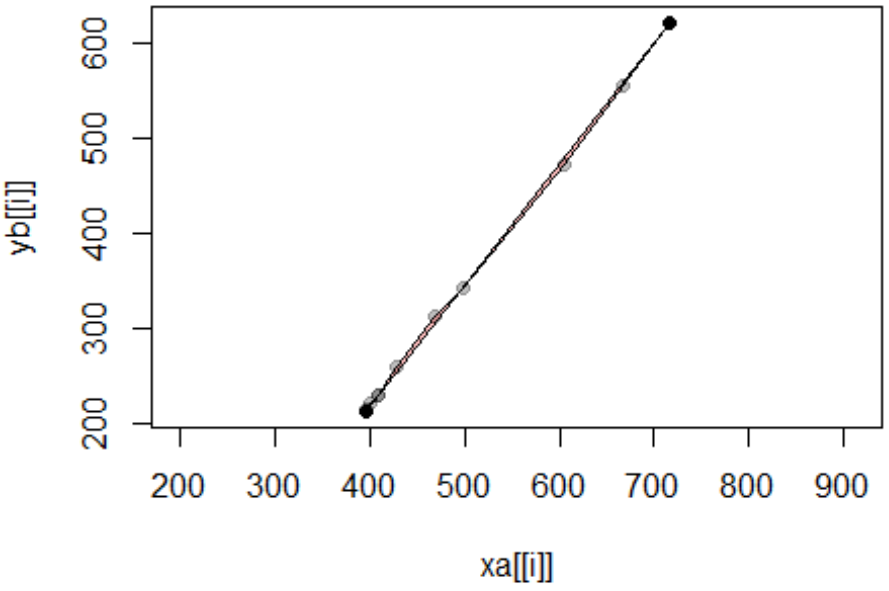

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

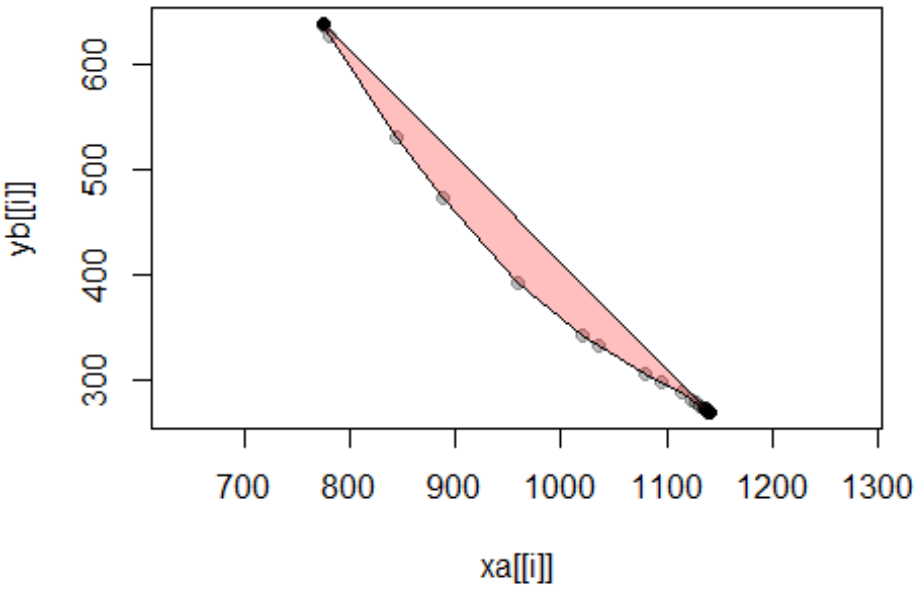

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

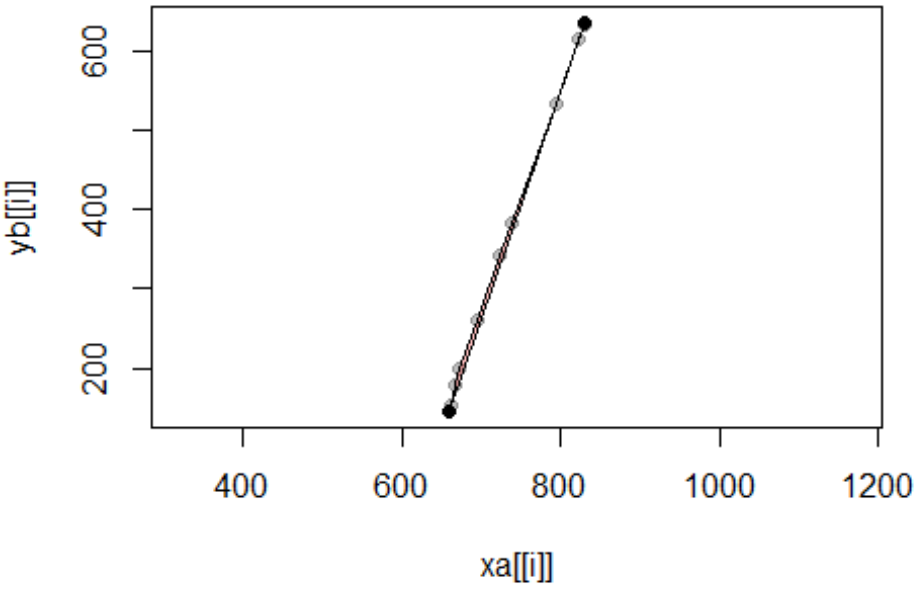

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

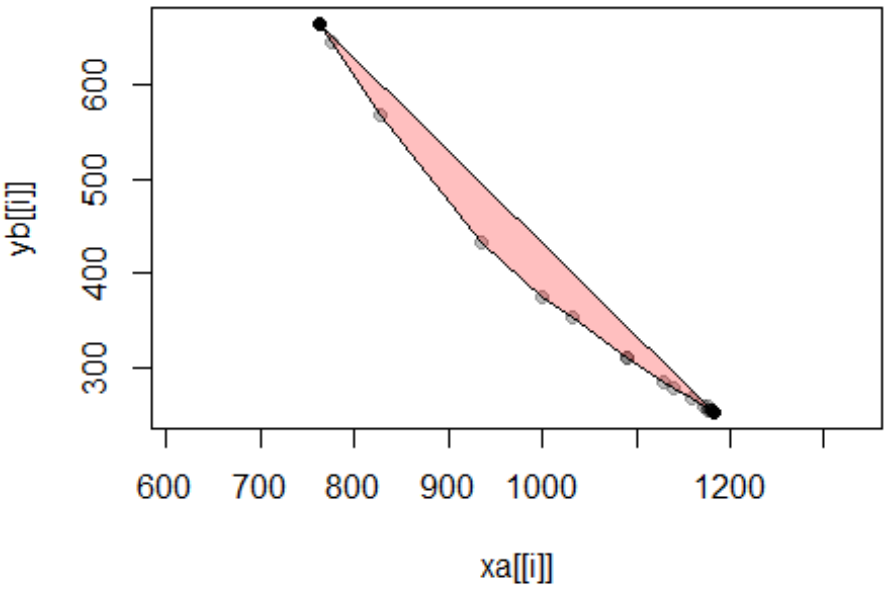

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

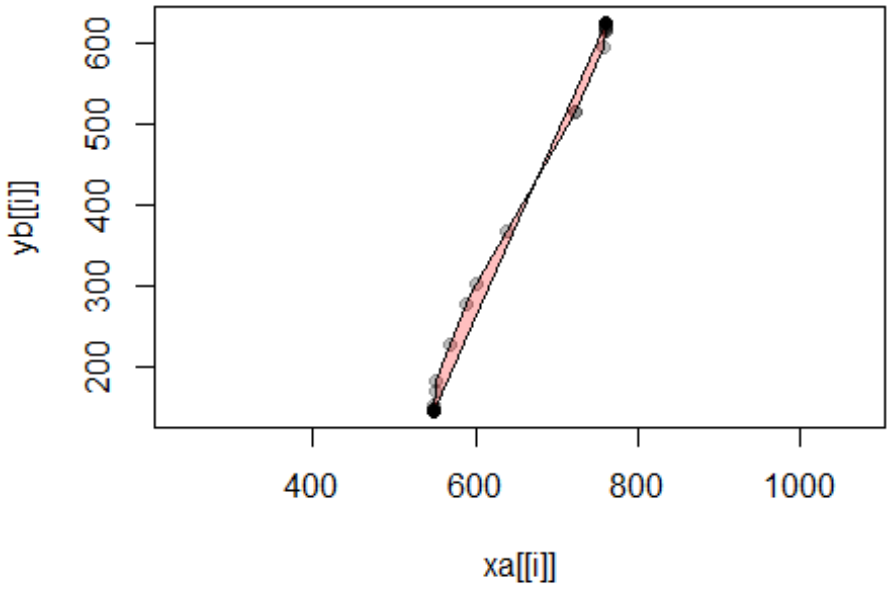

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

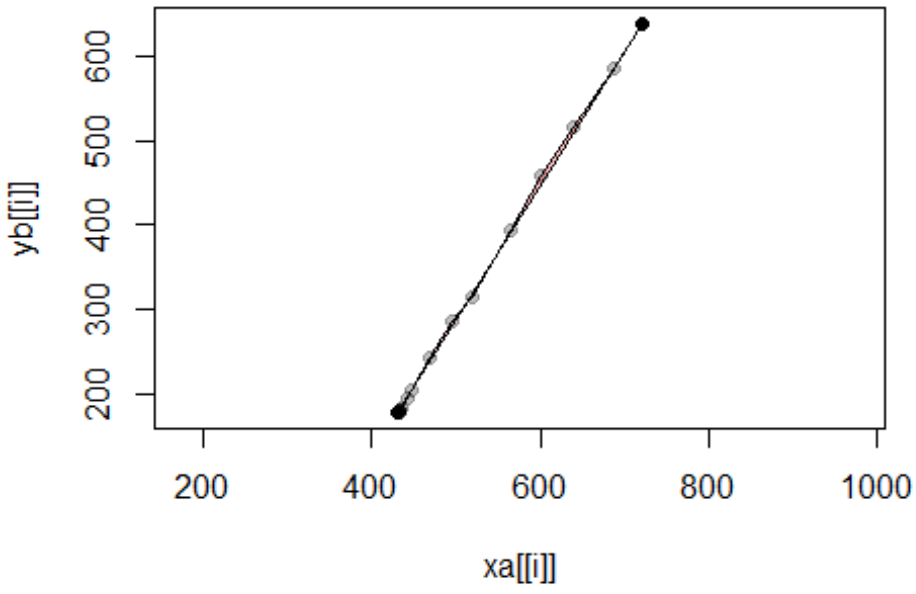

yb[[i]]
200
300
400
500
600
200
300
400
500
600
700
800
900
xa[[i]]

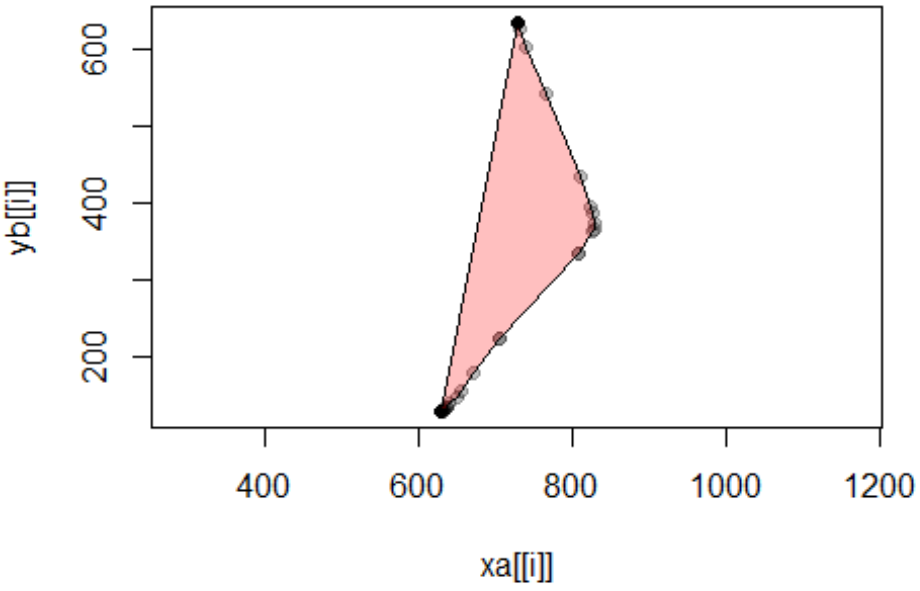

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

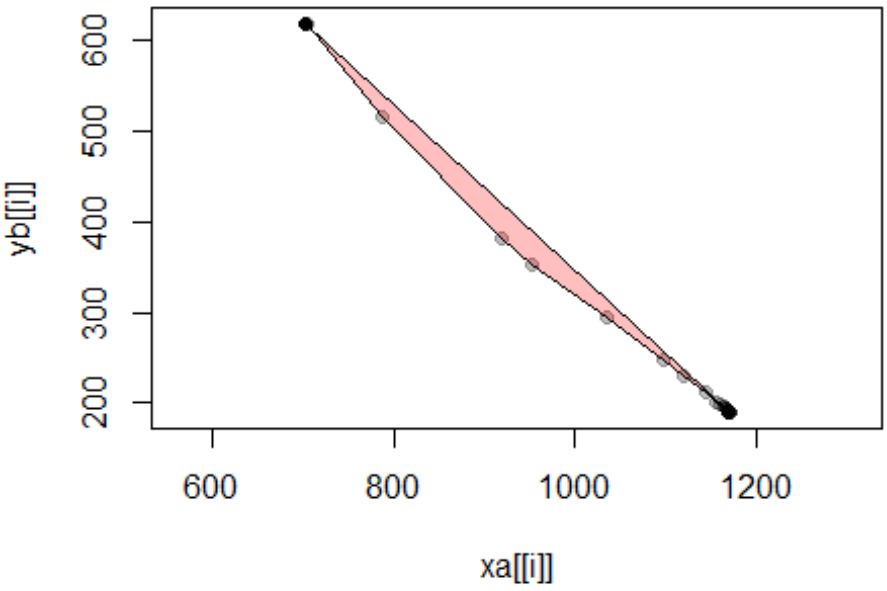

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

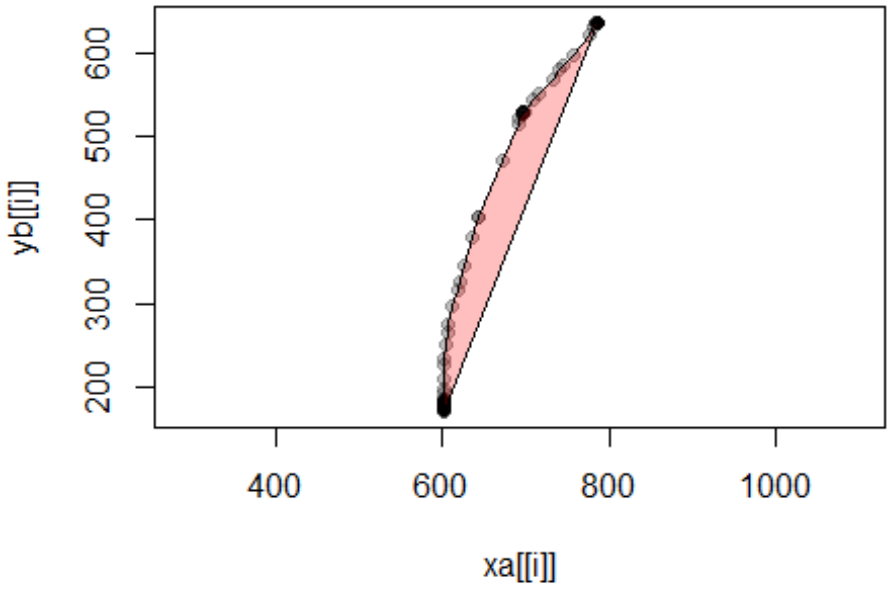

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

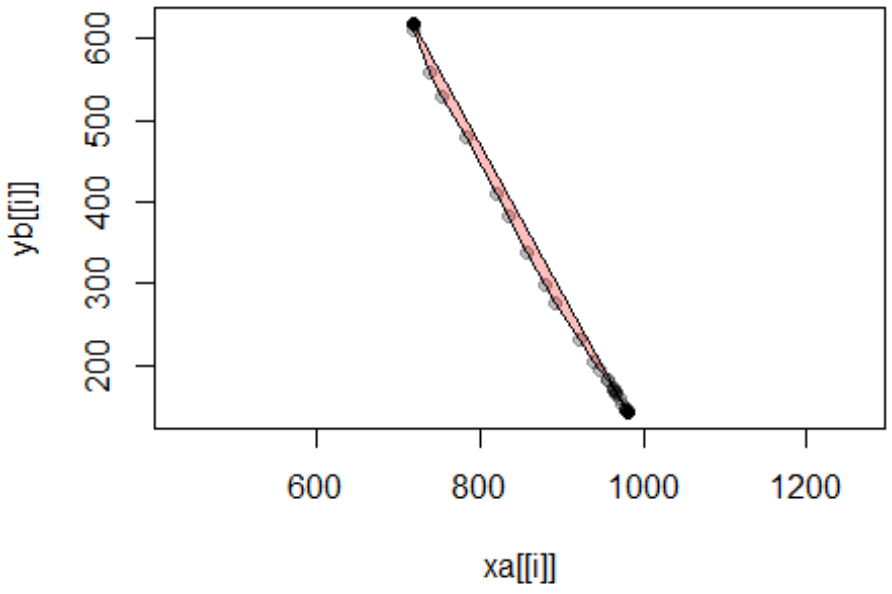

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

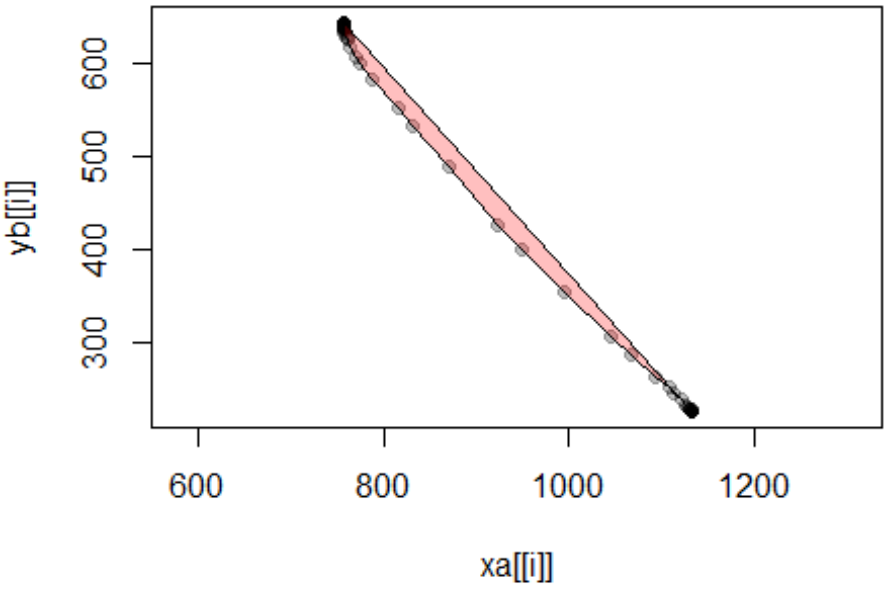

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

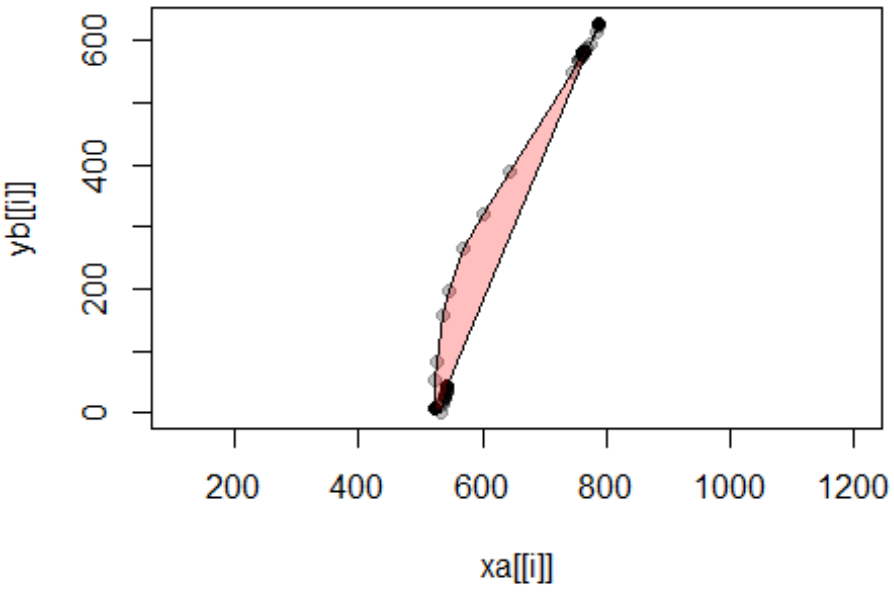

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

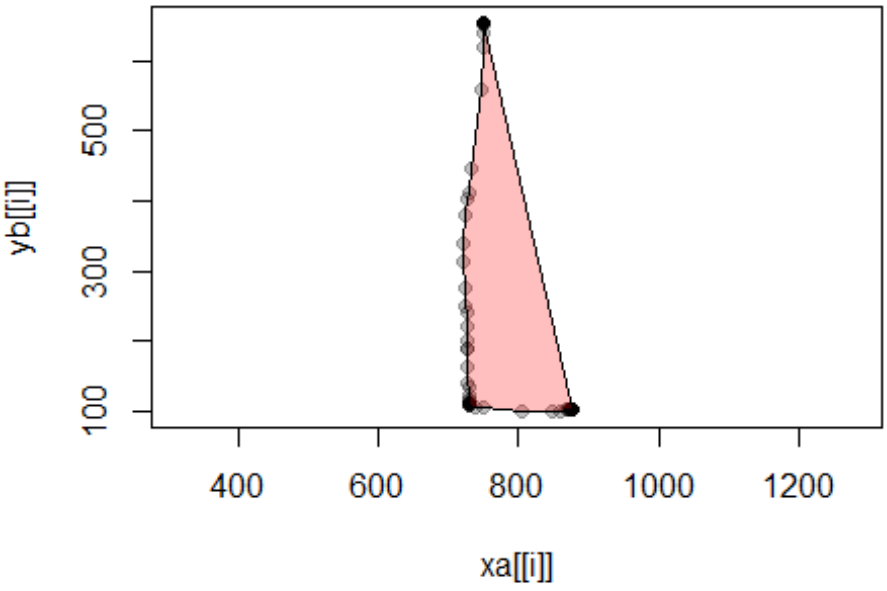

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

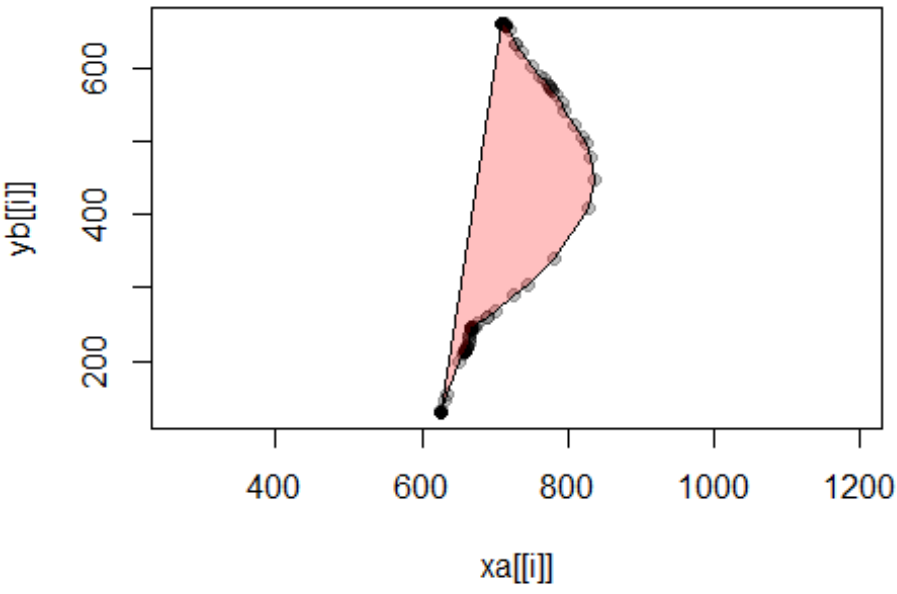

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

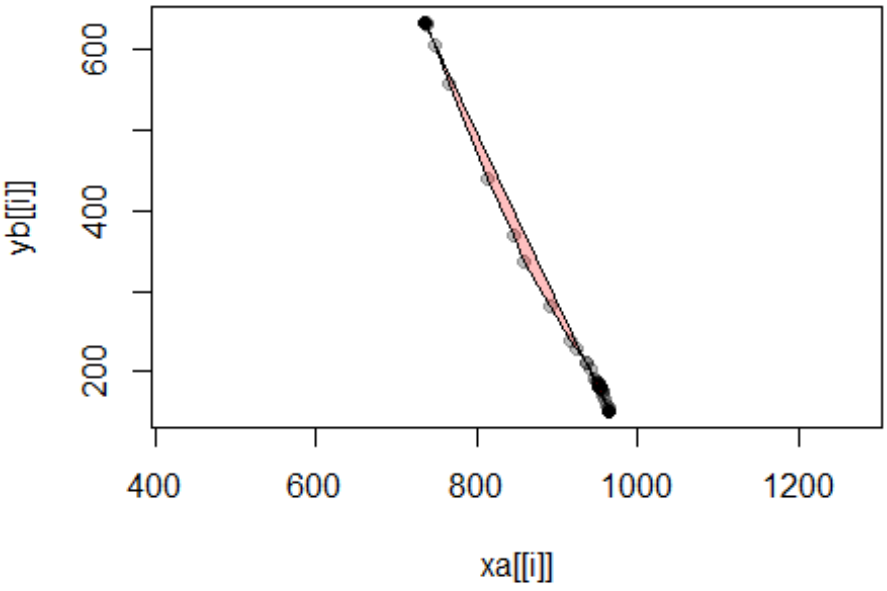

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

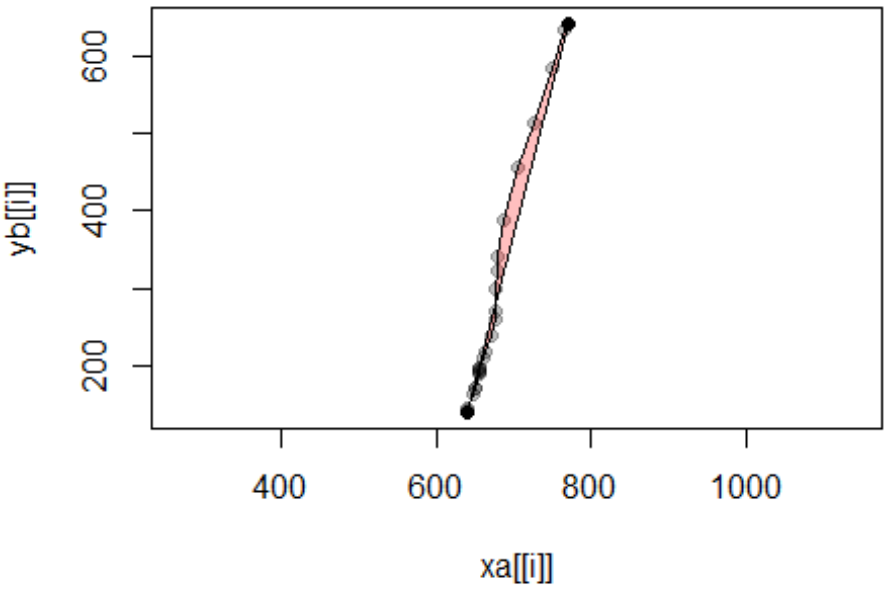

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

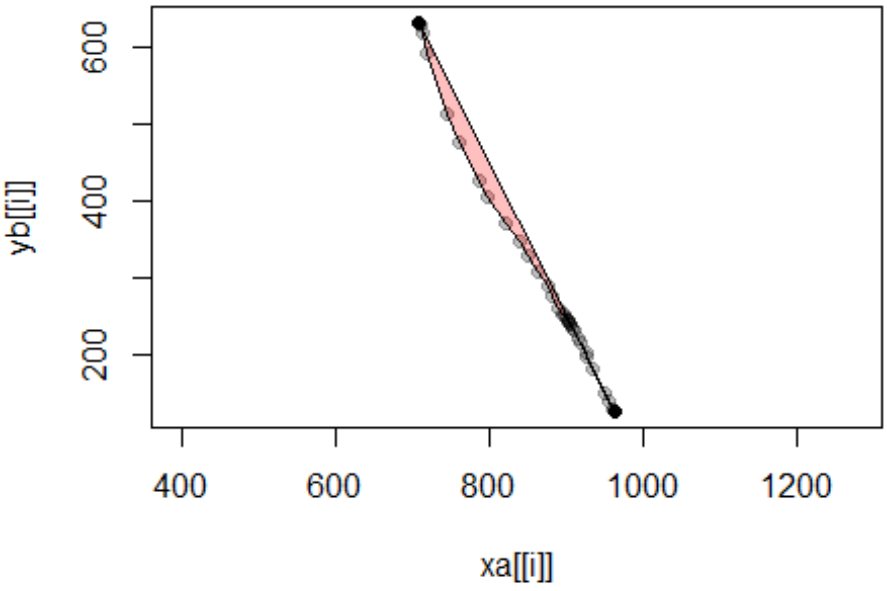

yb[[i]]
0
200
400
600
200
400
600
800
1000
1200
xa[[i]]

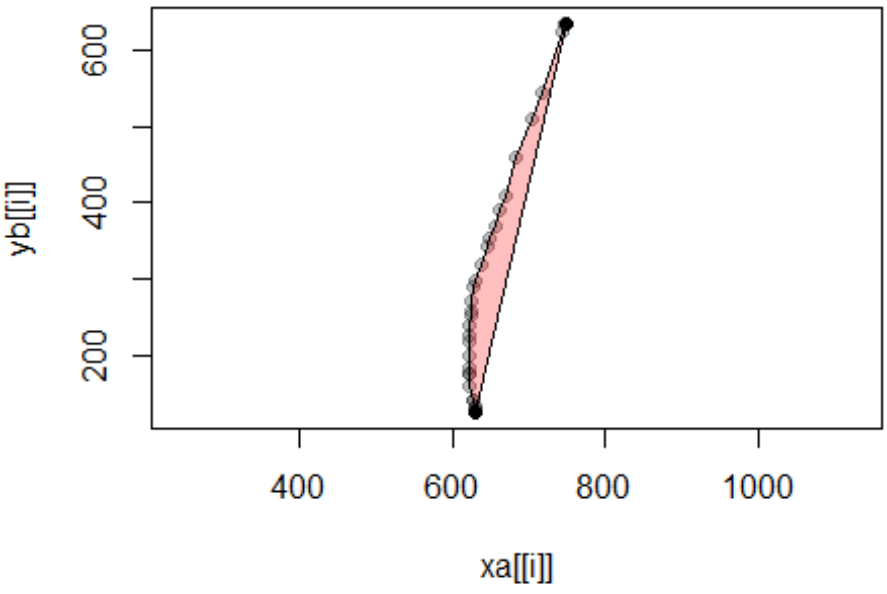

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

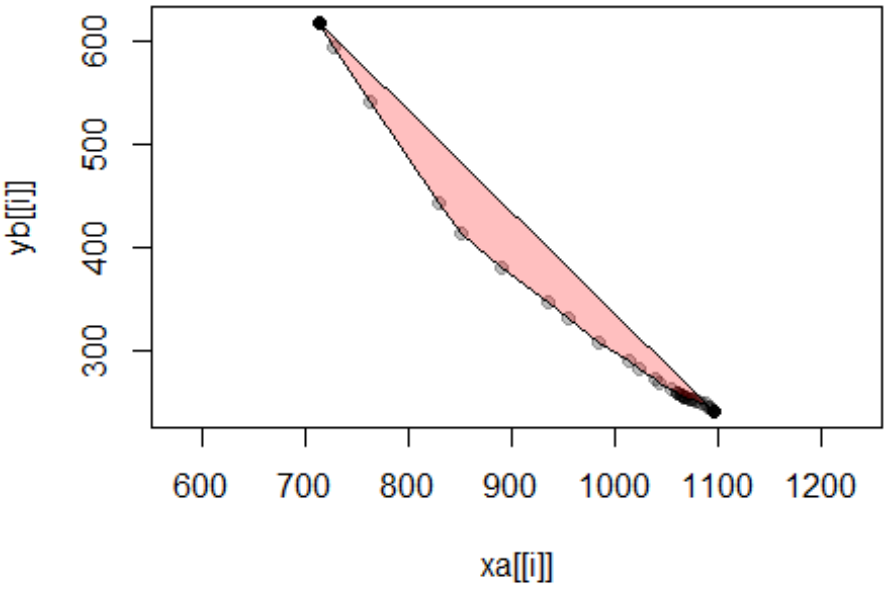

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

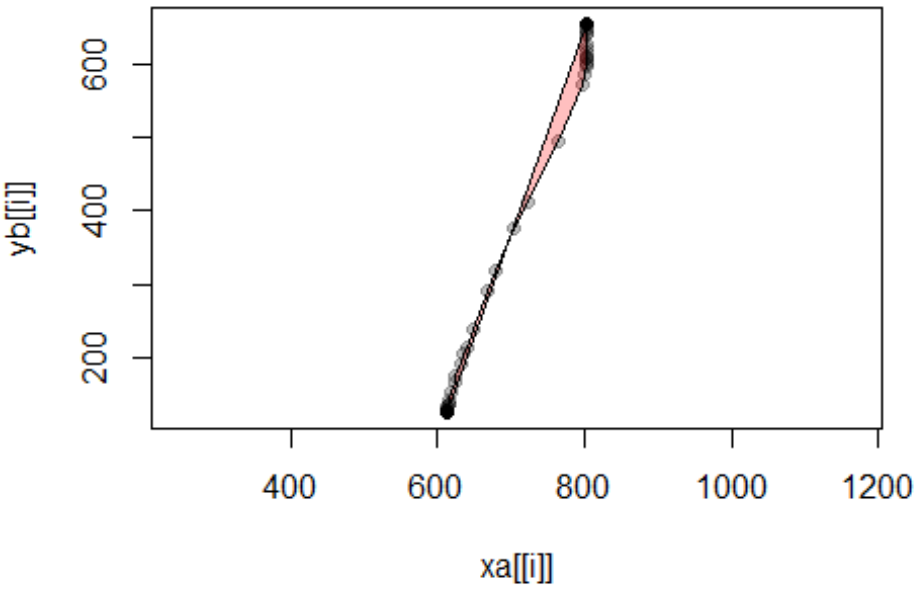

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

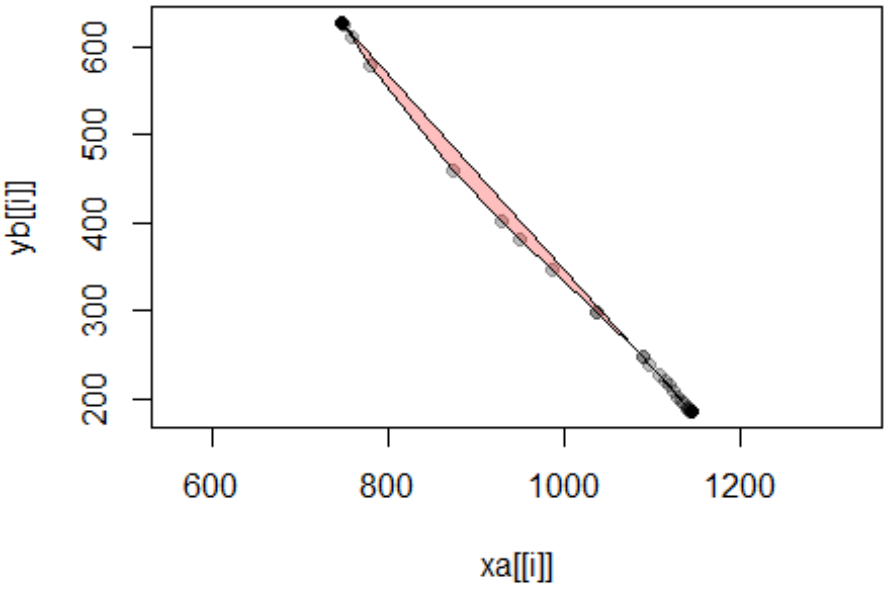

yb[[i]]
600
400
200
400
600
800
1000
xa[[i]]

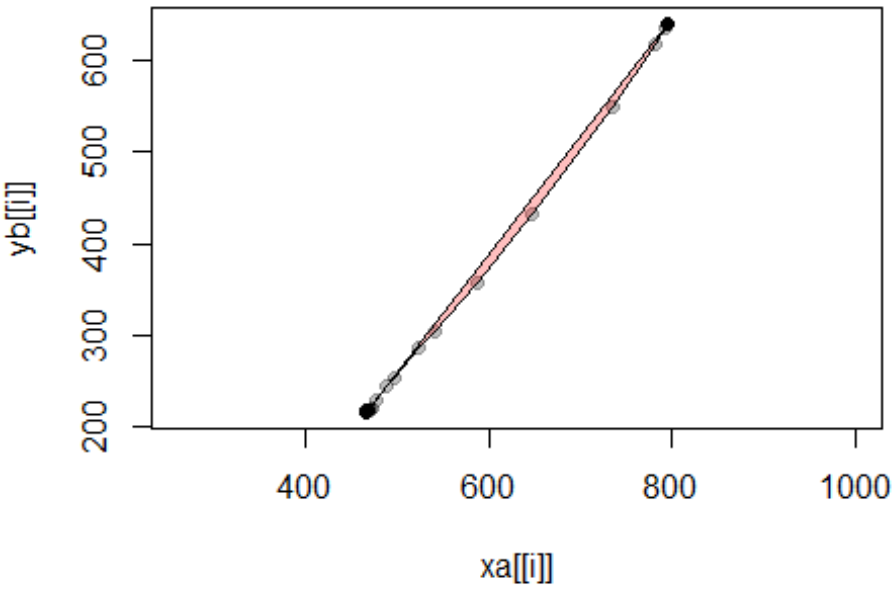

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

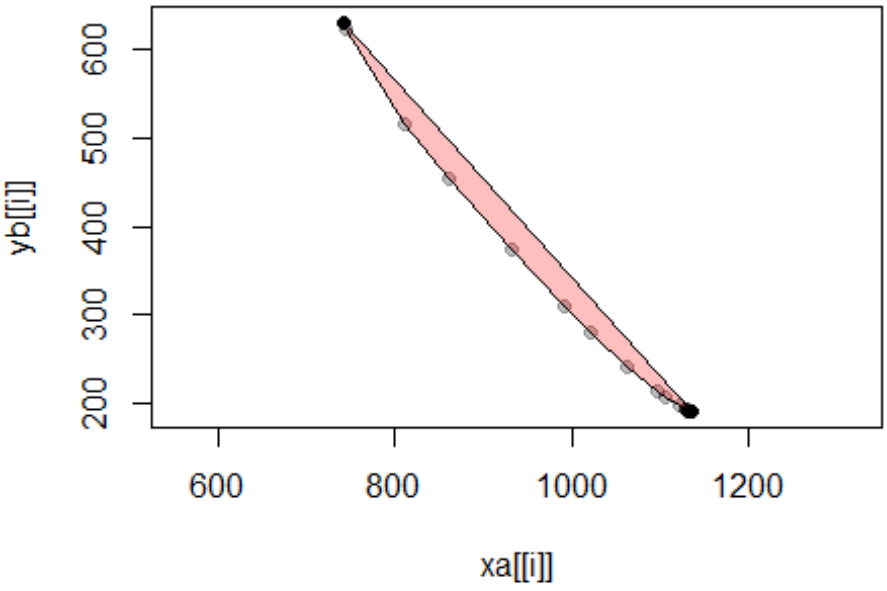

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

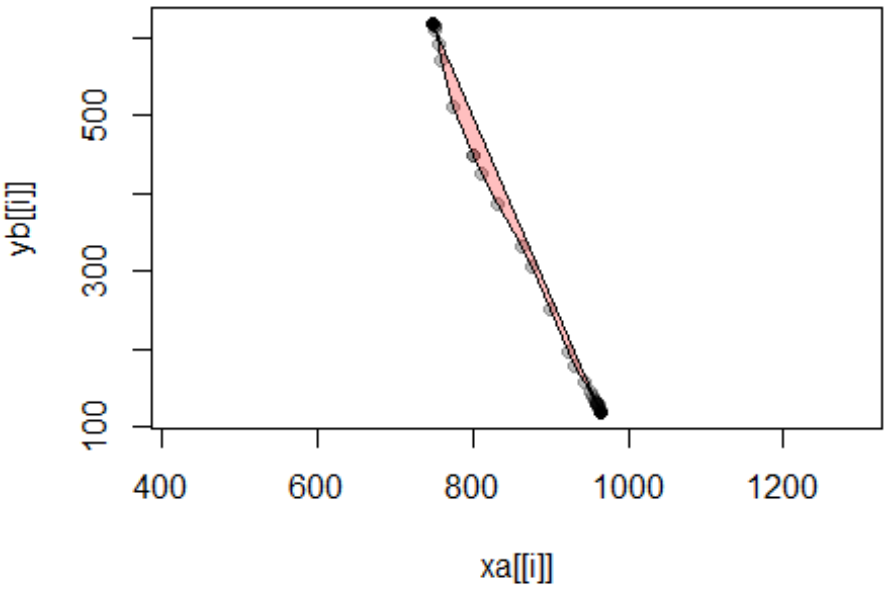

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
xa[[i]]

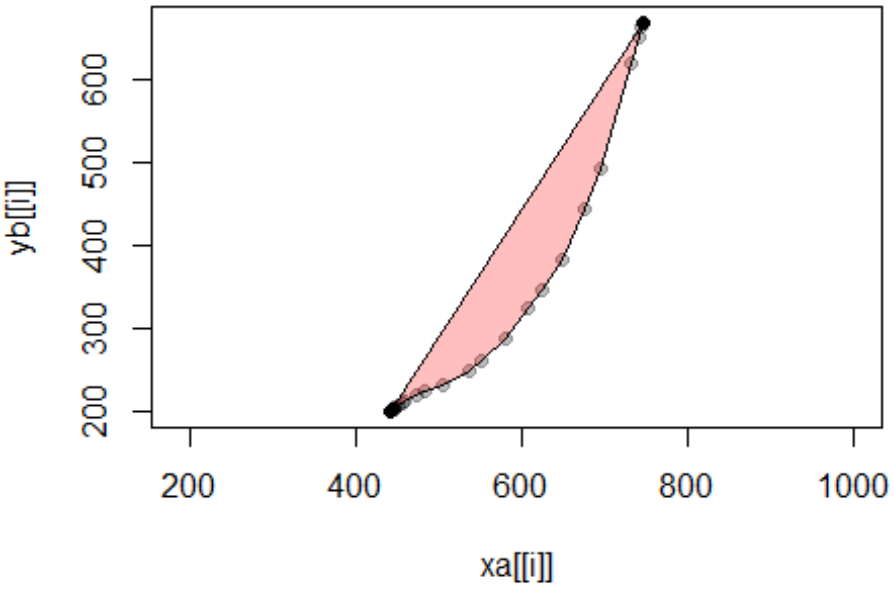

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

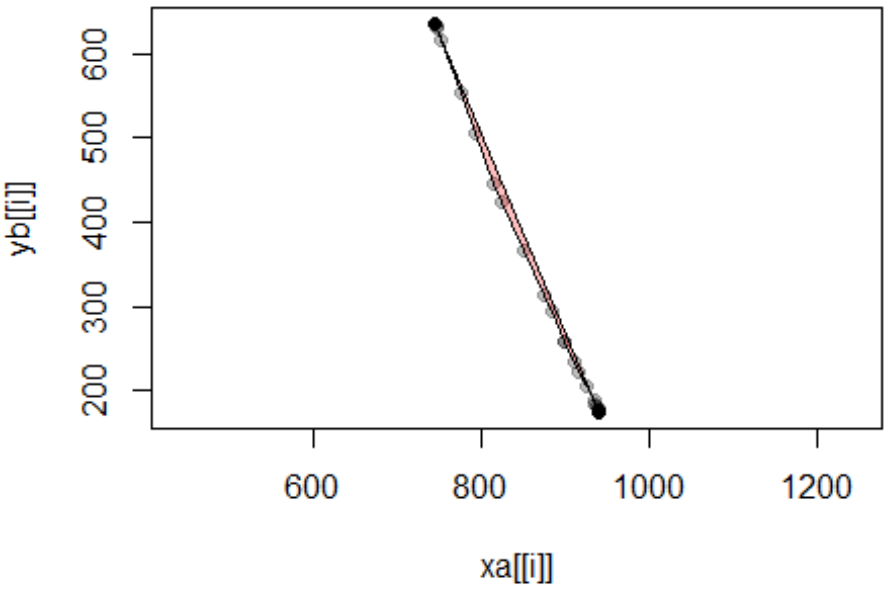

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

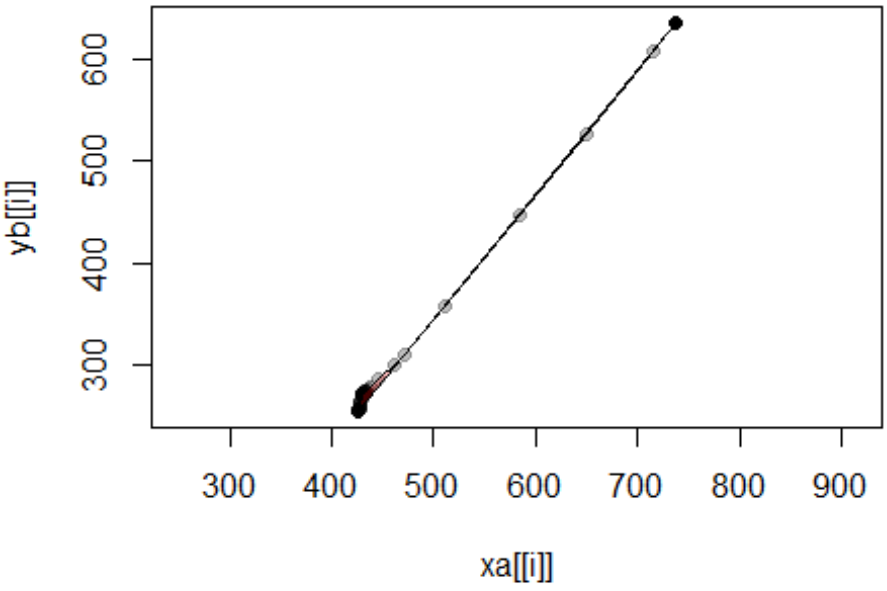

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

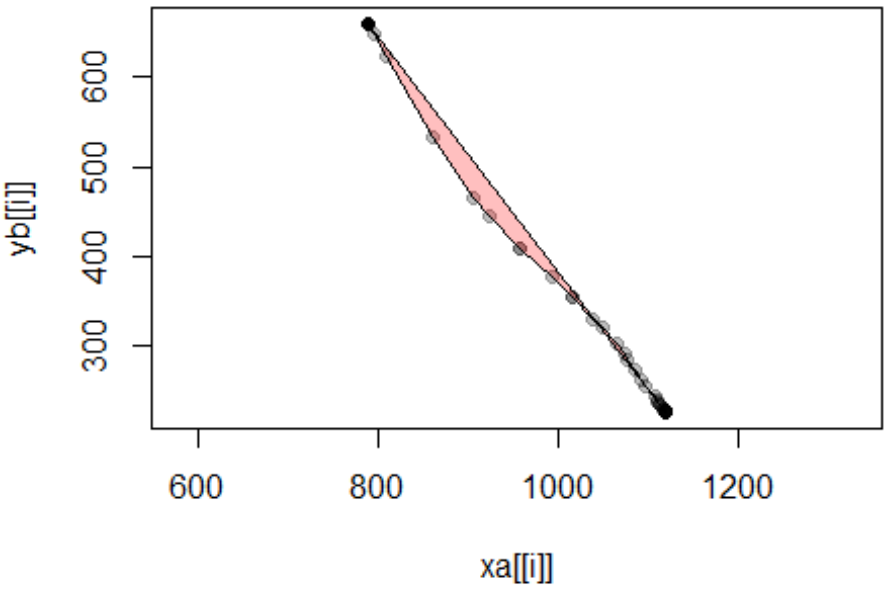

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

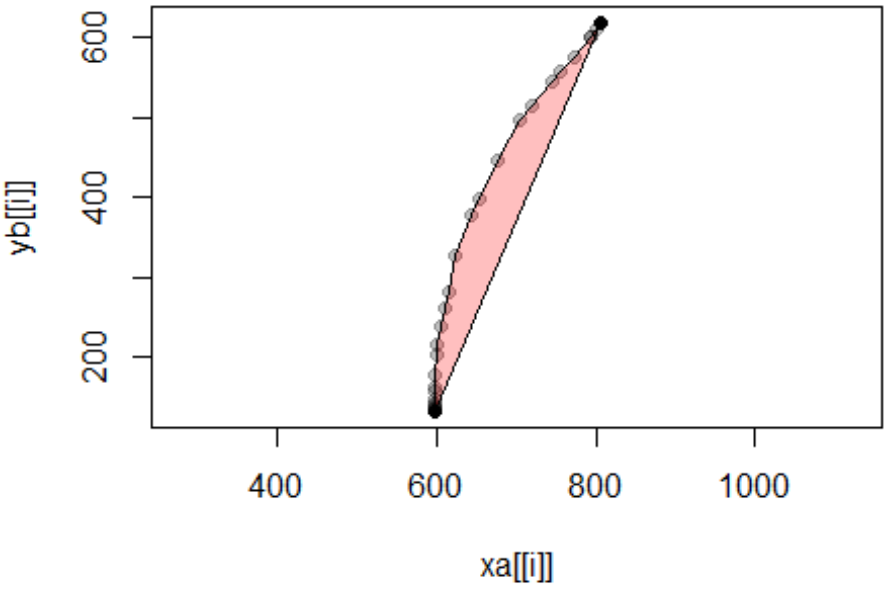

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

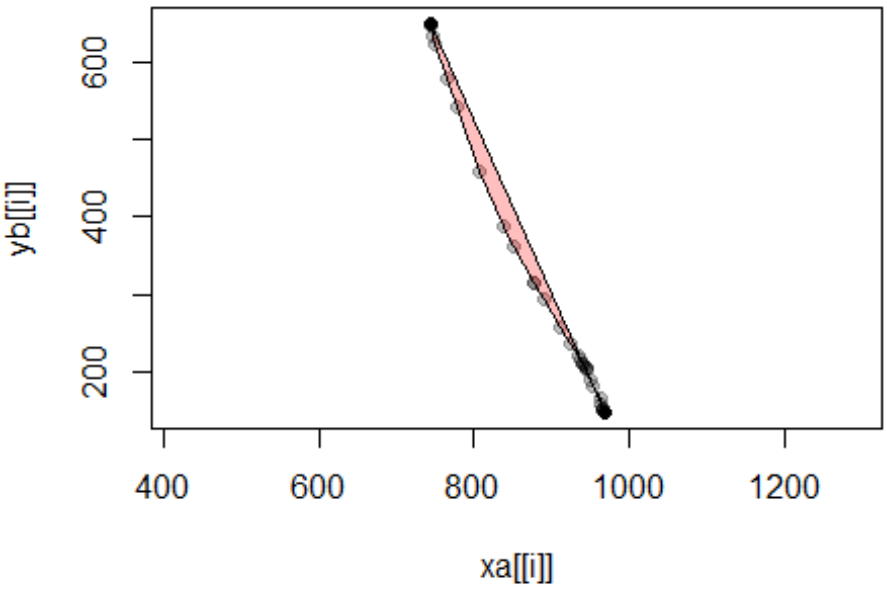

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

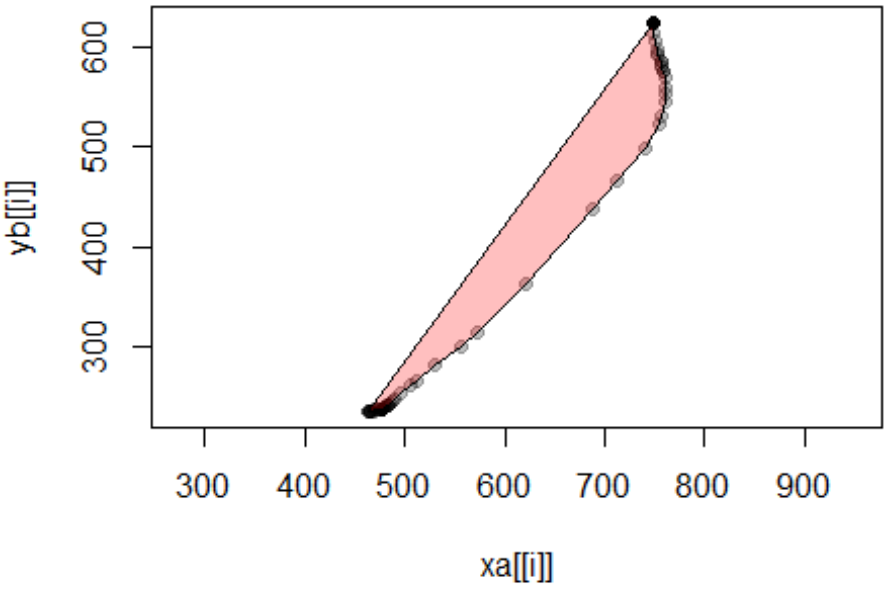

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

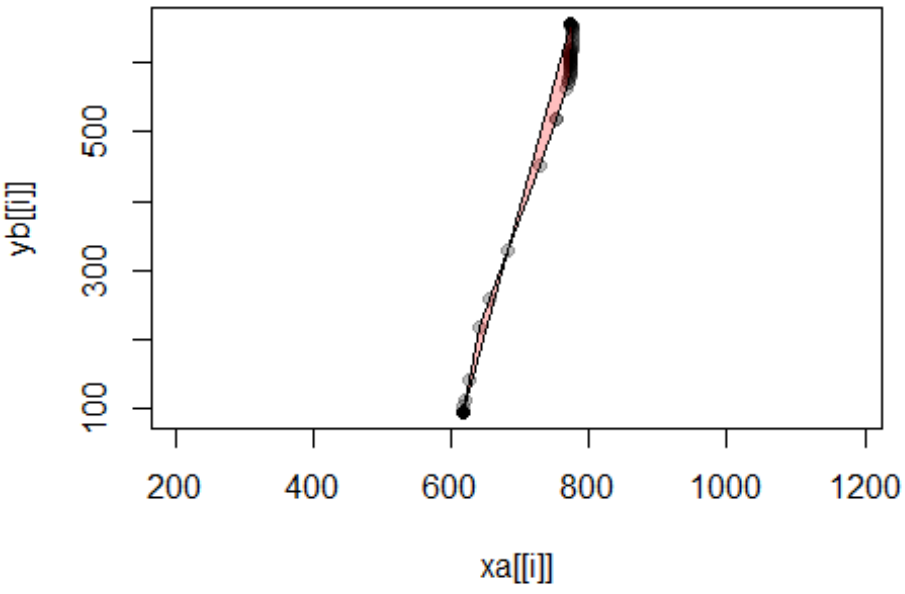

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

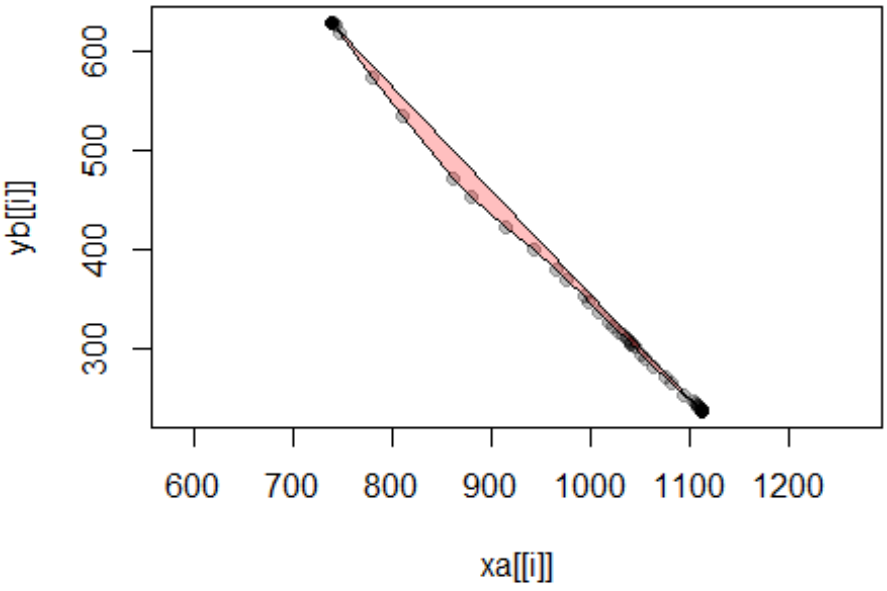

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

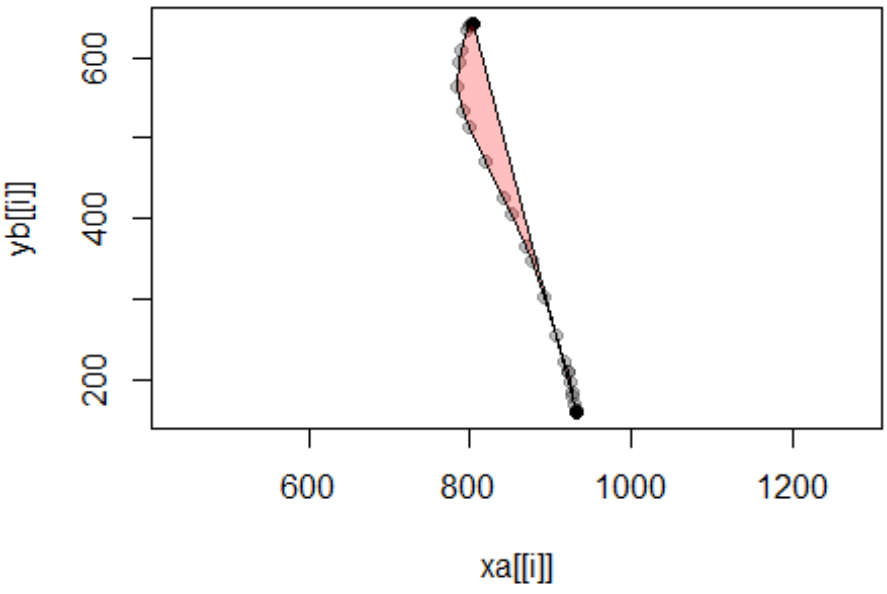

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

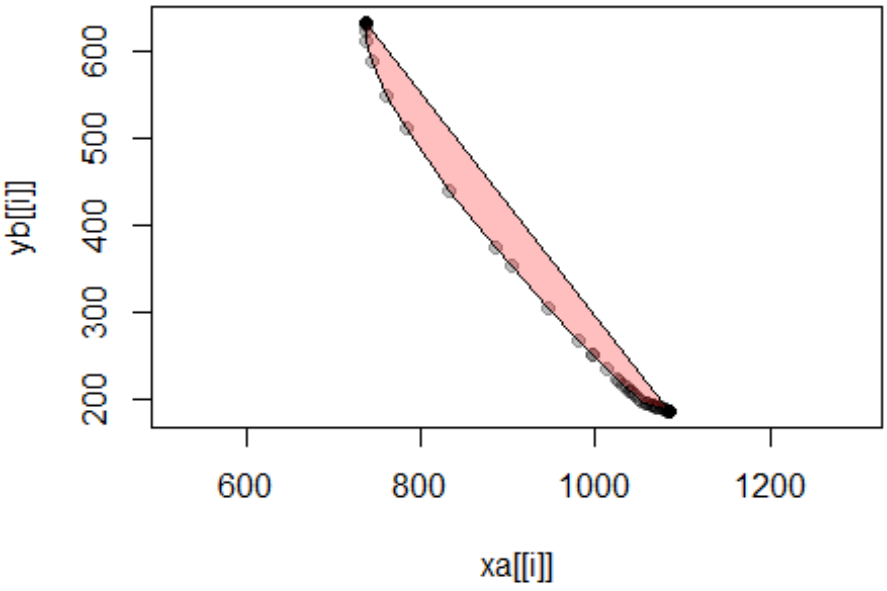

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

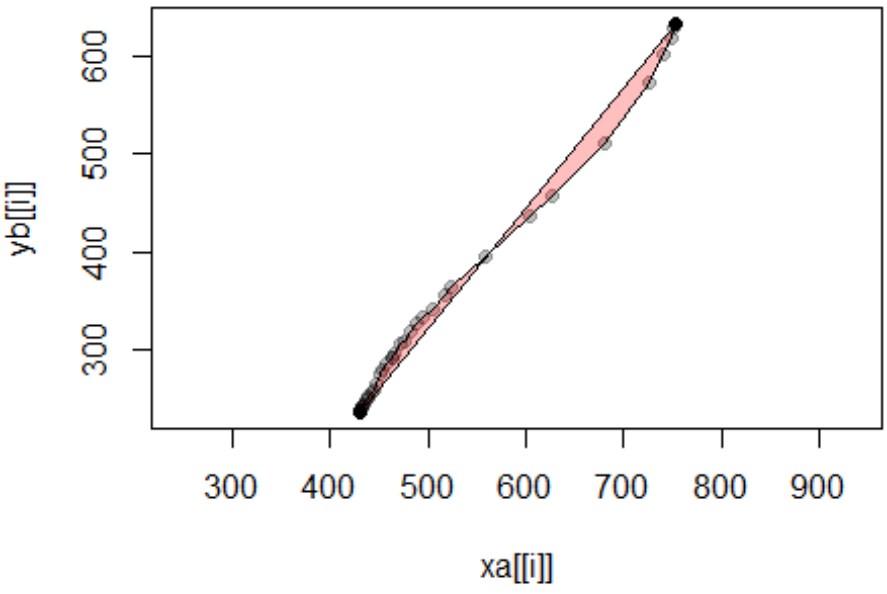

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

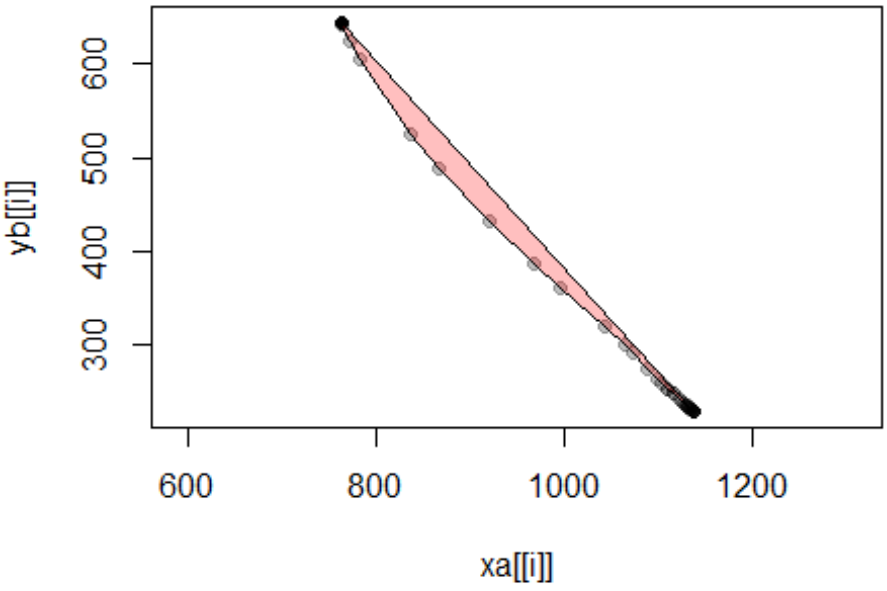

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

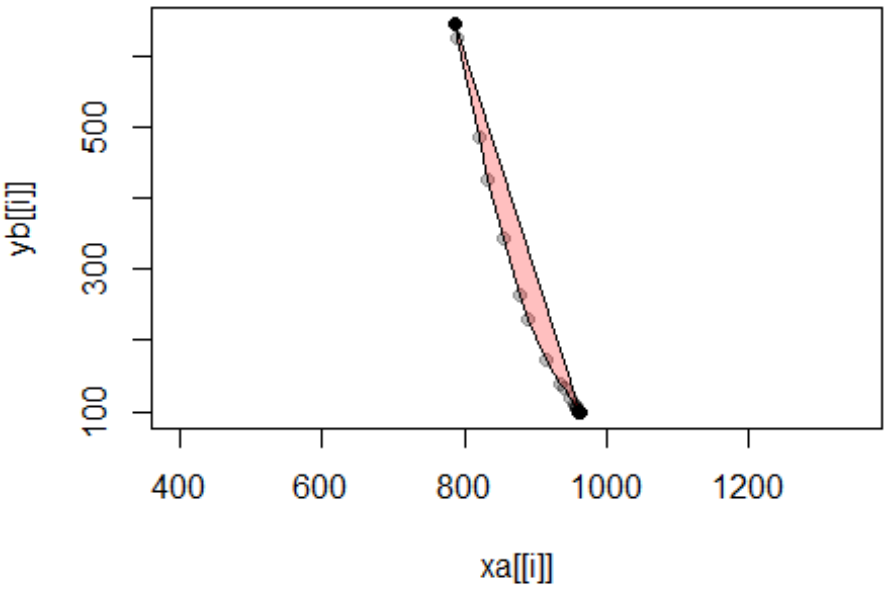

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
xa[[i]]

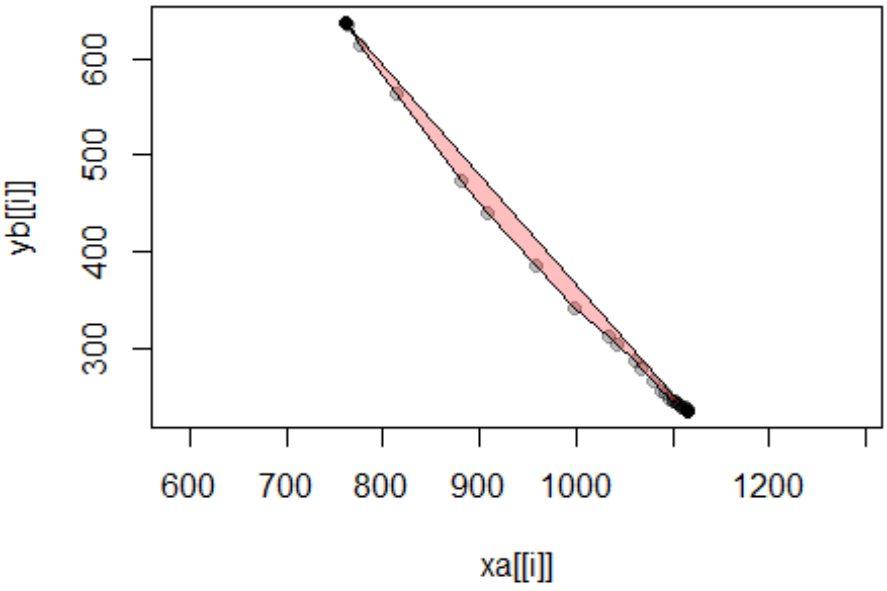

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

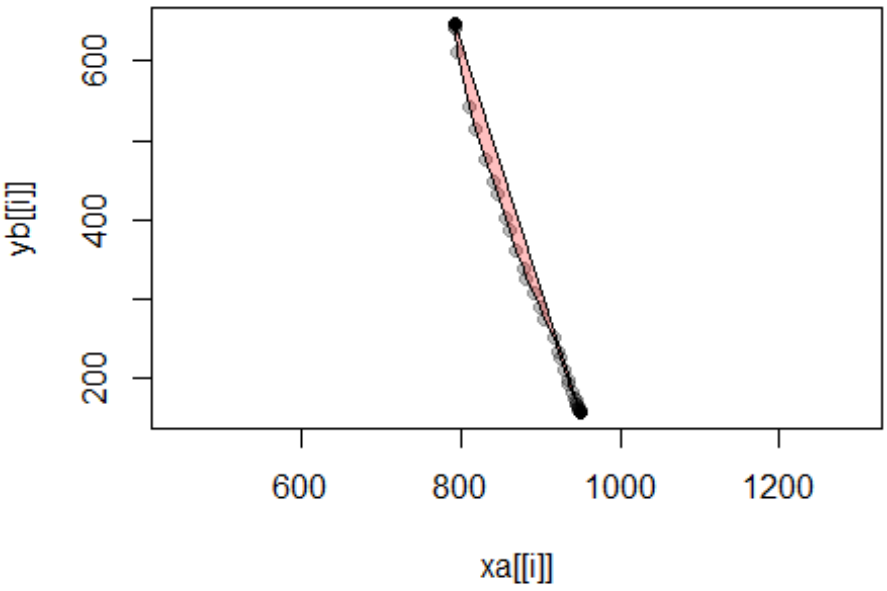

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

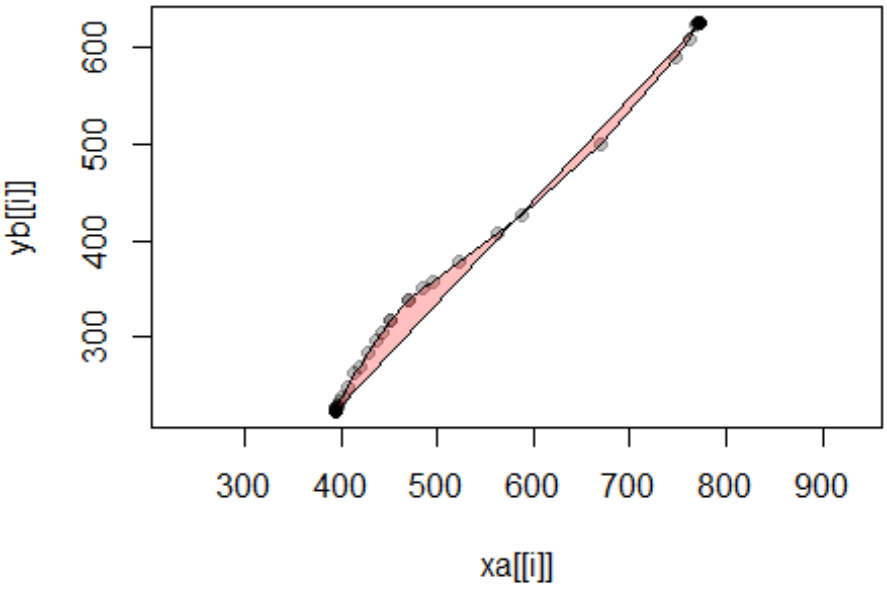

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

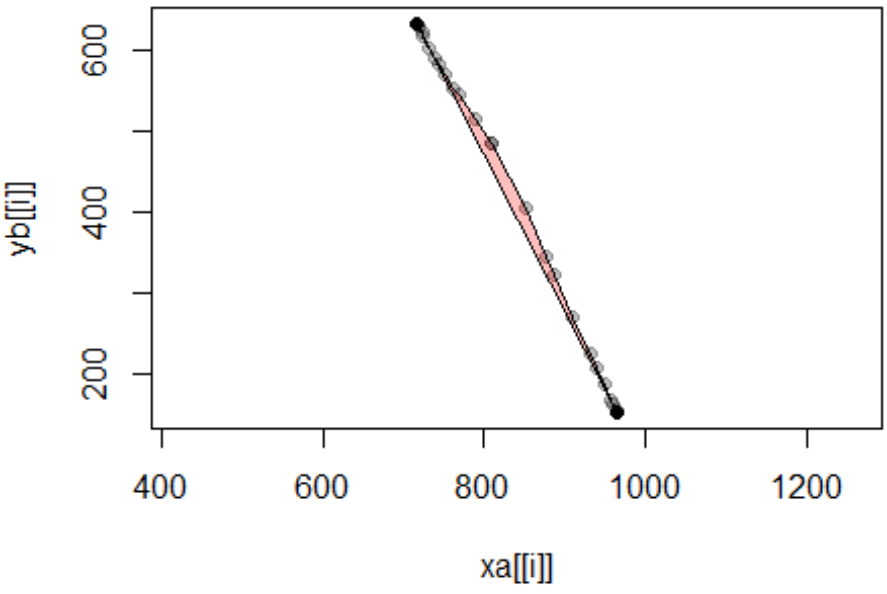

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

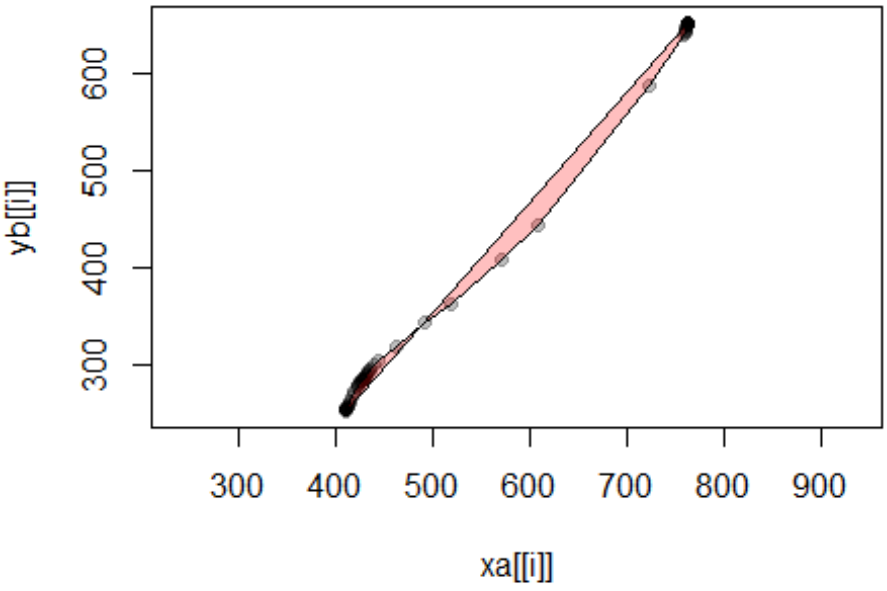

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

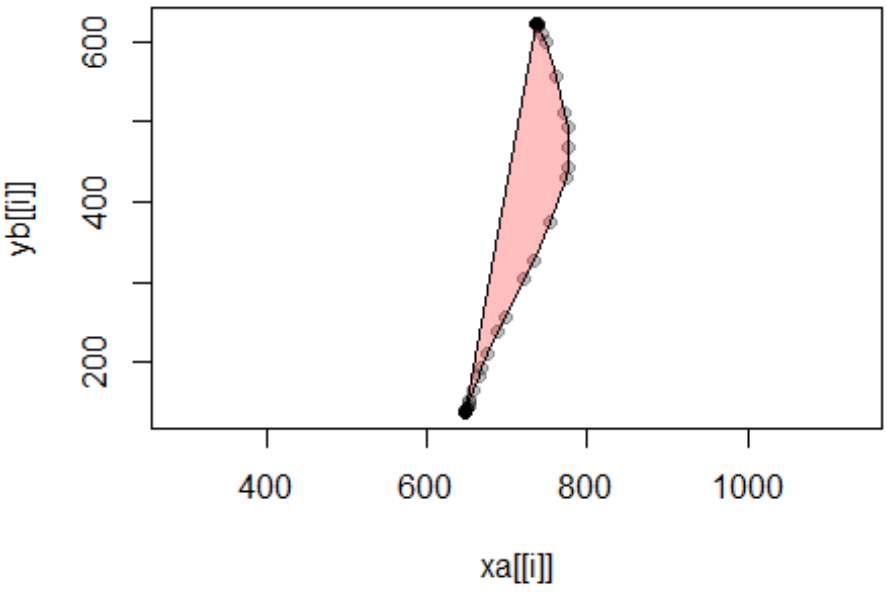

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

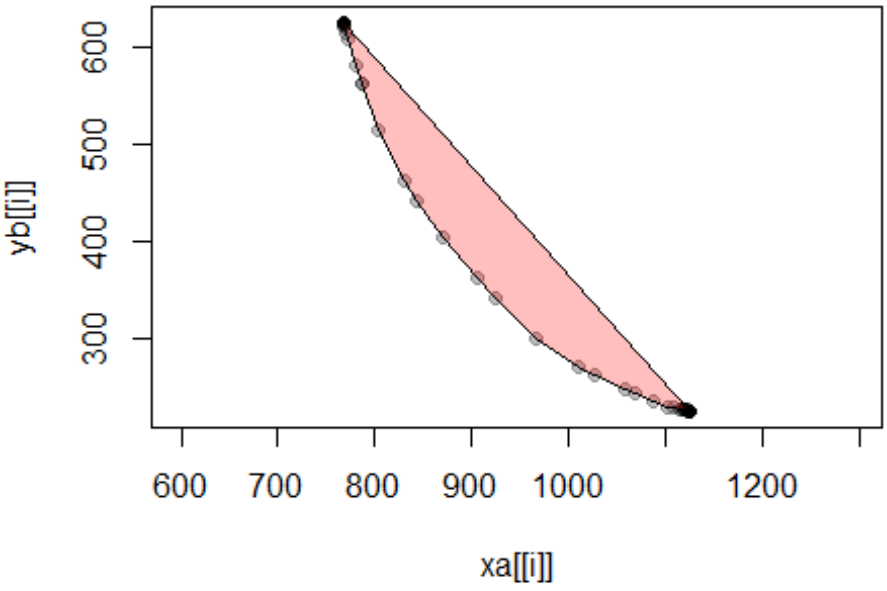

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

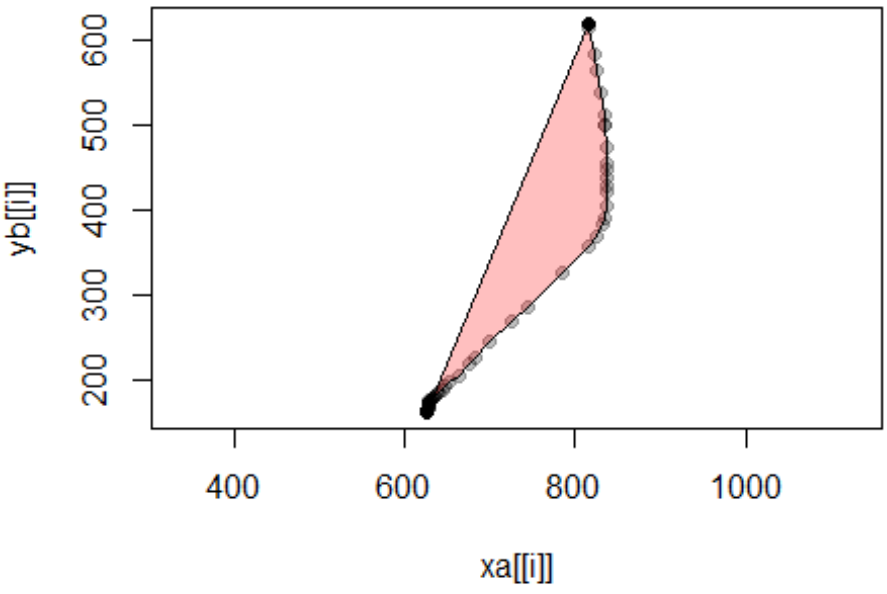

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

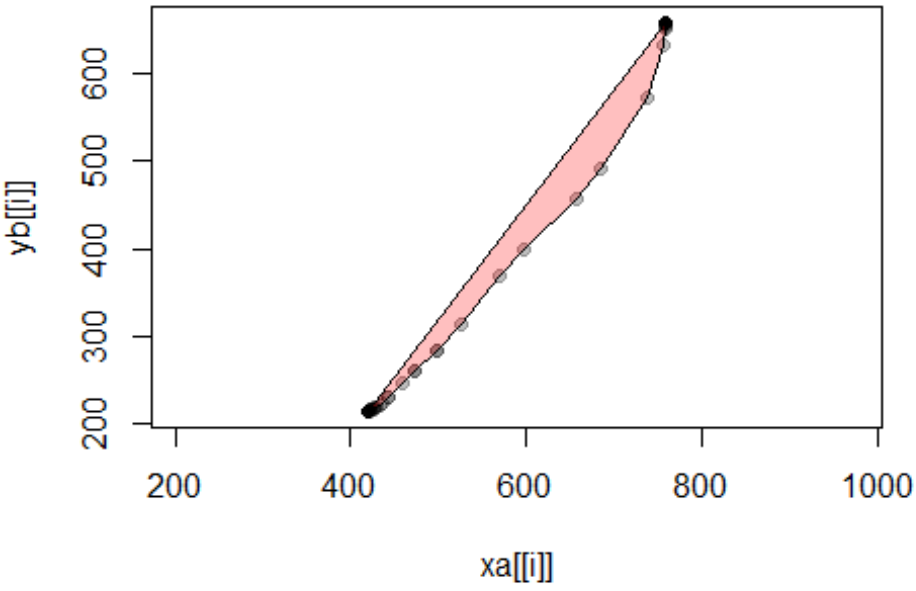

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

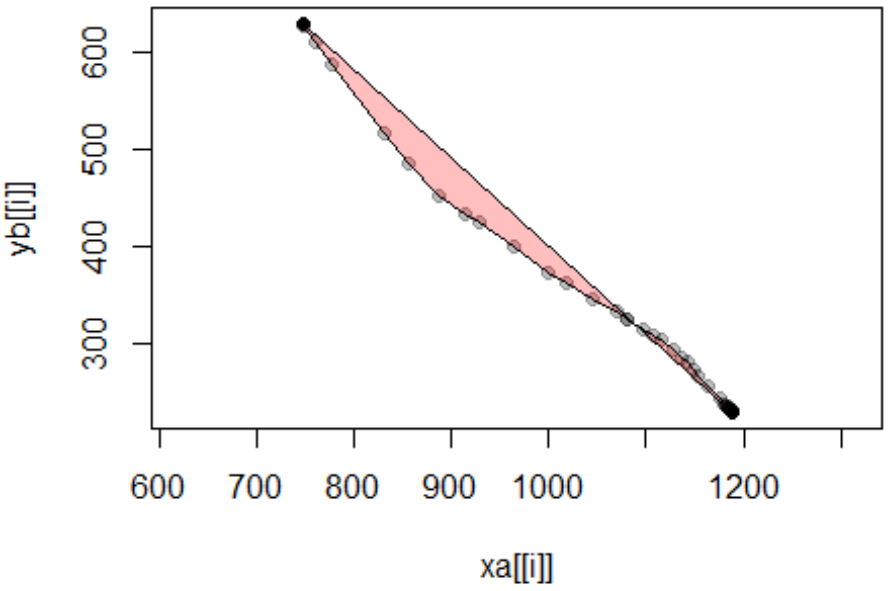

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

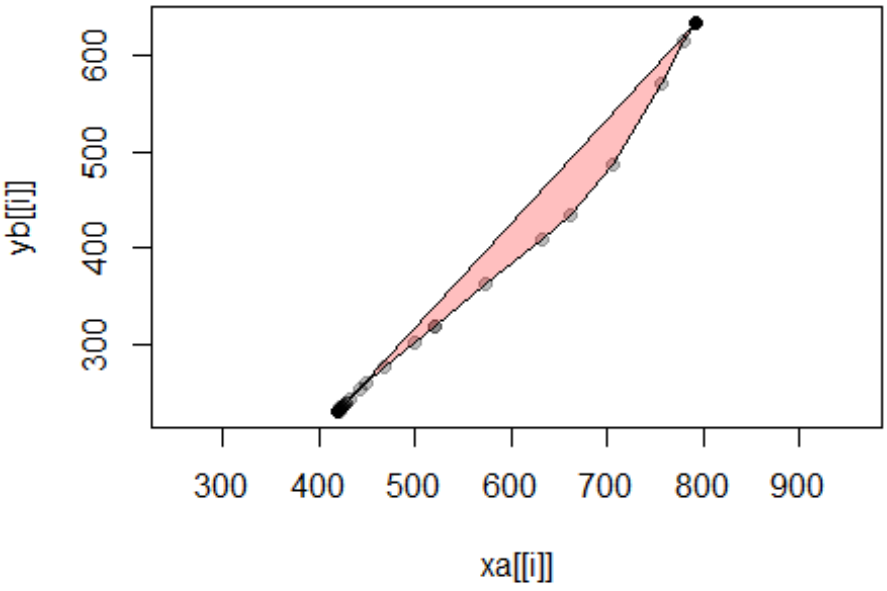

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

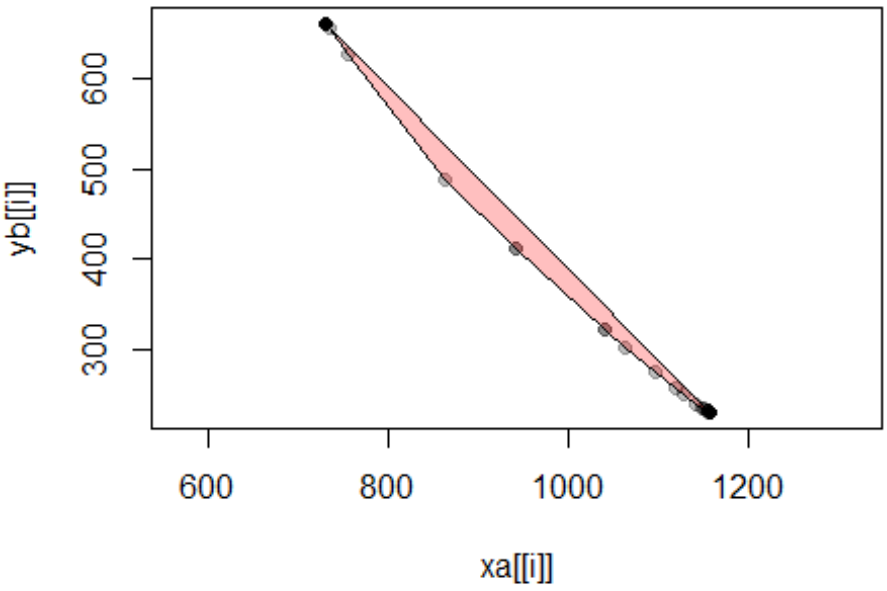

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

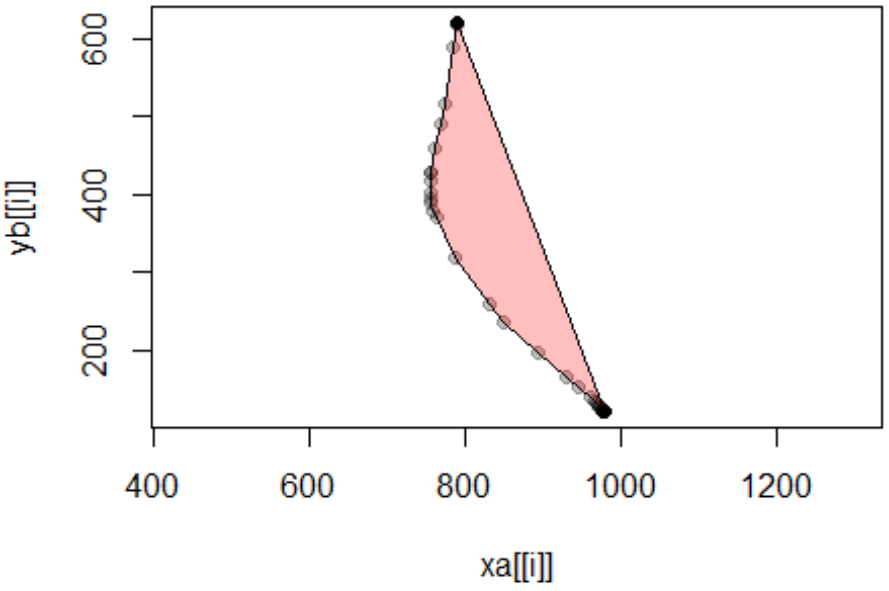

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

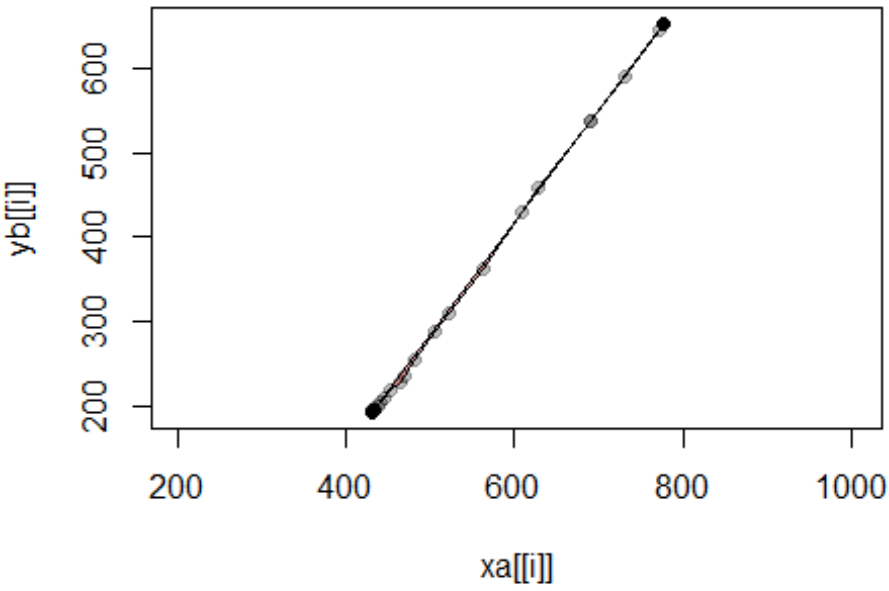

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

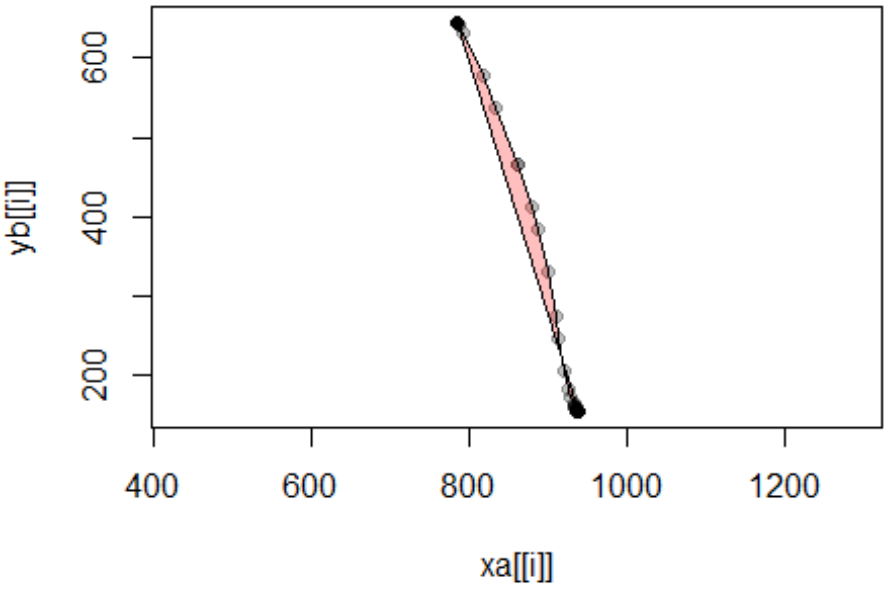

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

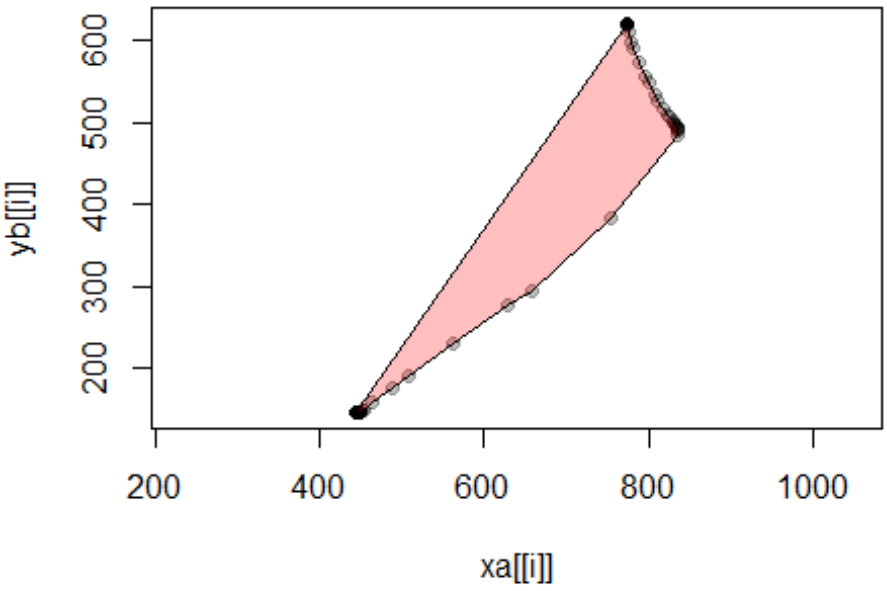

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

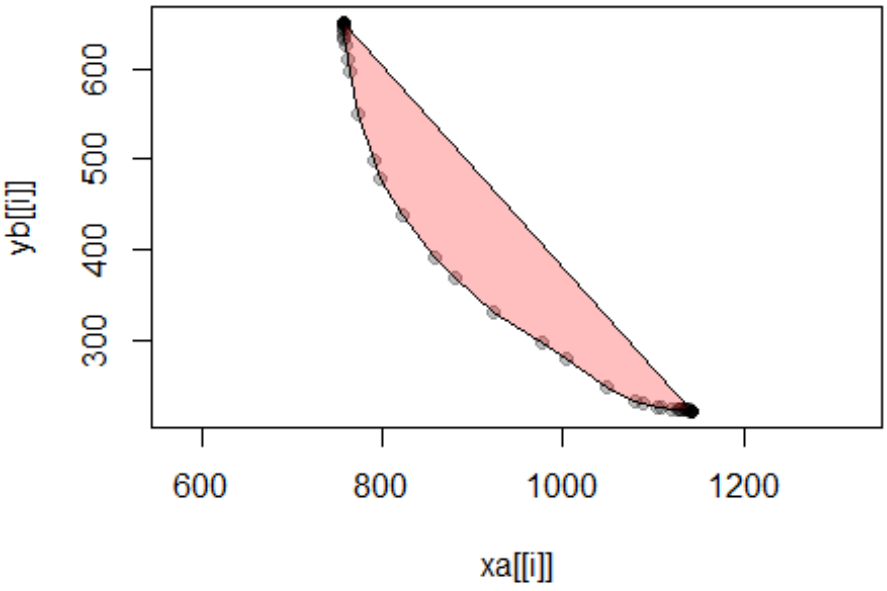

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

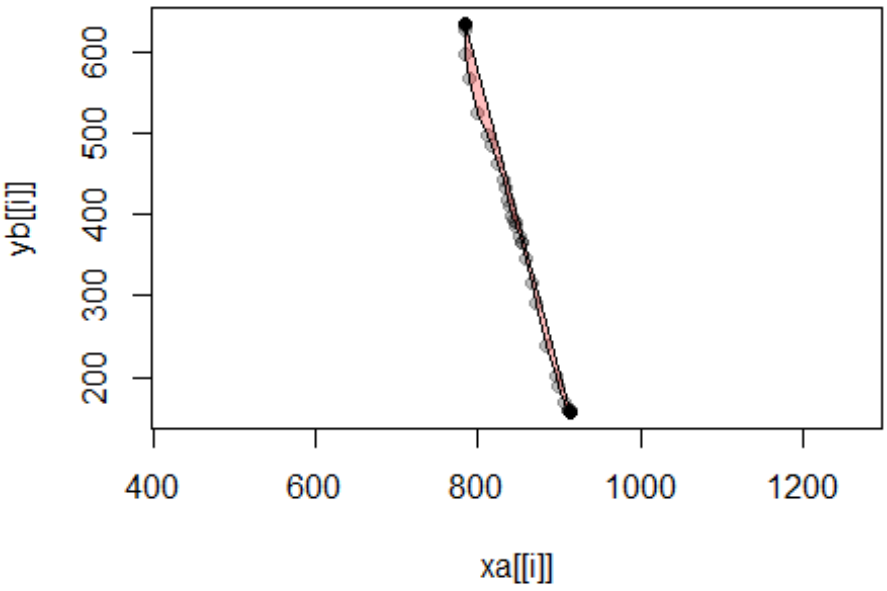

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

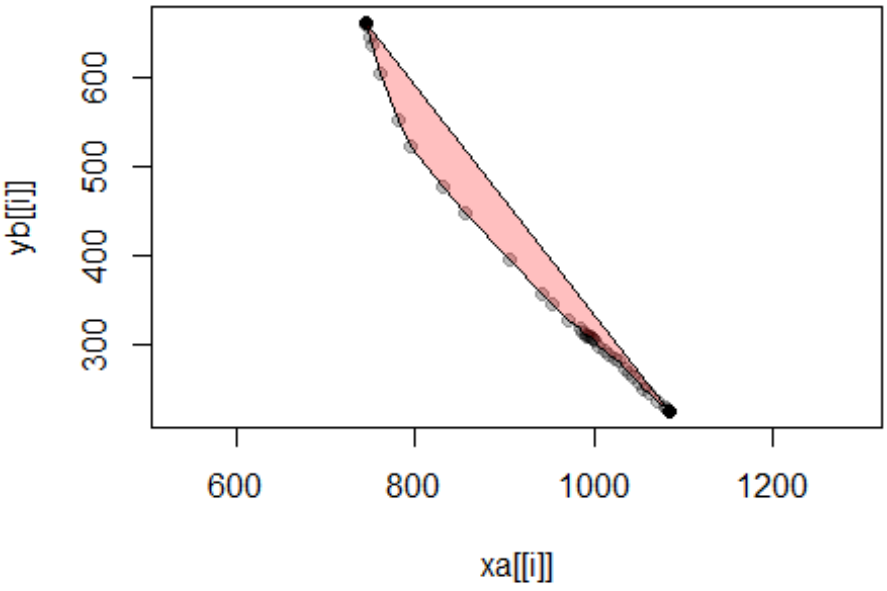

yb[[i]]
600
500
400
300
200
200
400
600
800
1000
xa[[i]]

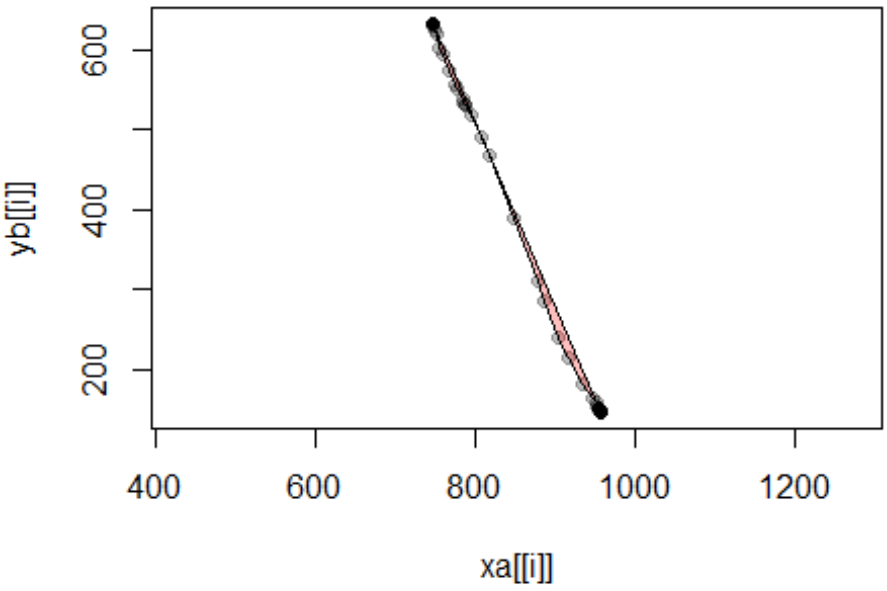

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

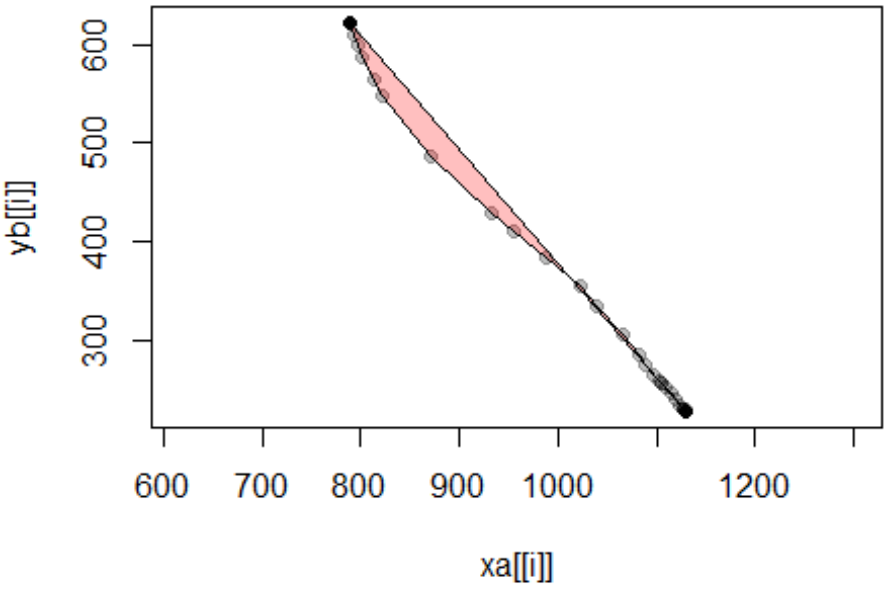

yb[[i]]
600
500
400
300
200
200
400
600
800
1000
xa[[i]]

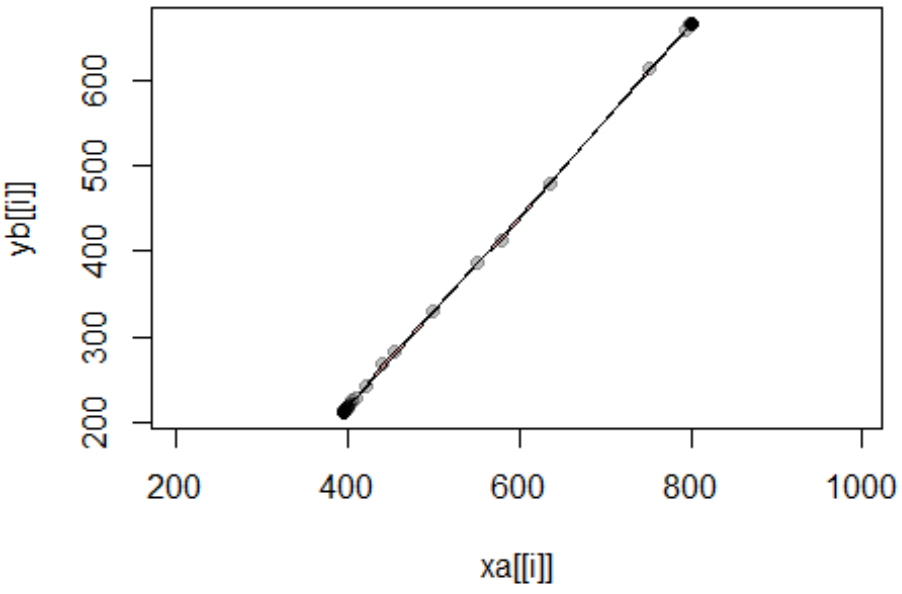

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

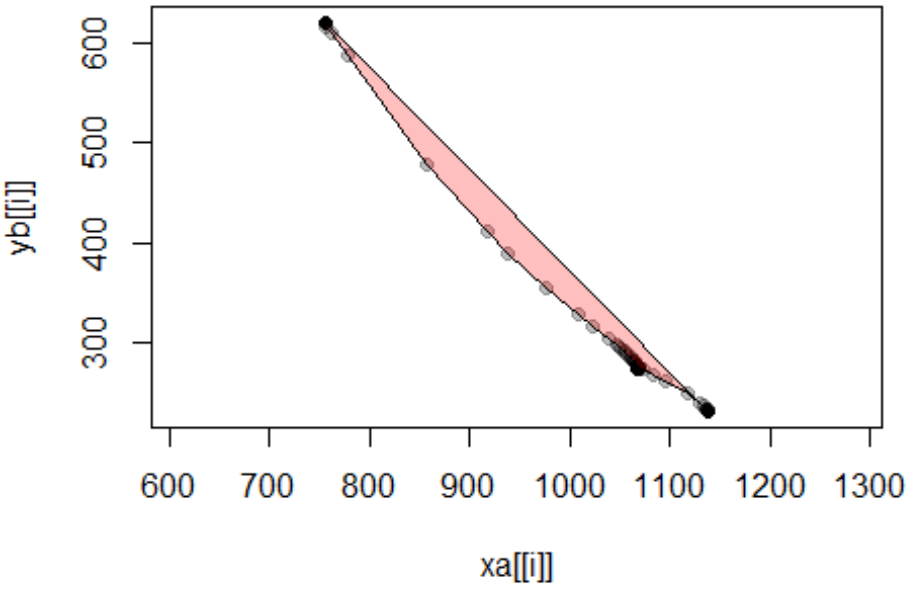

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

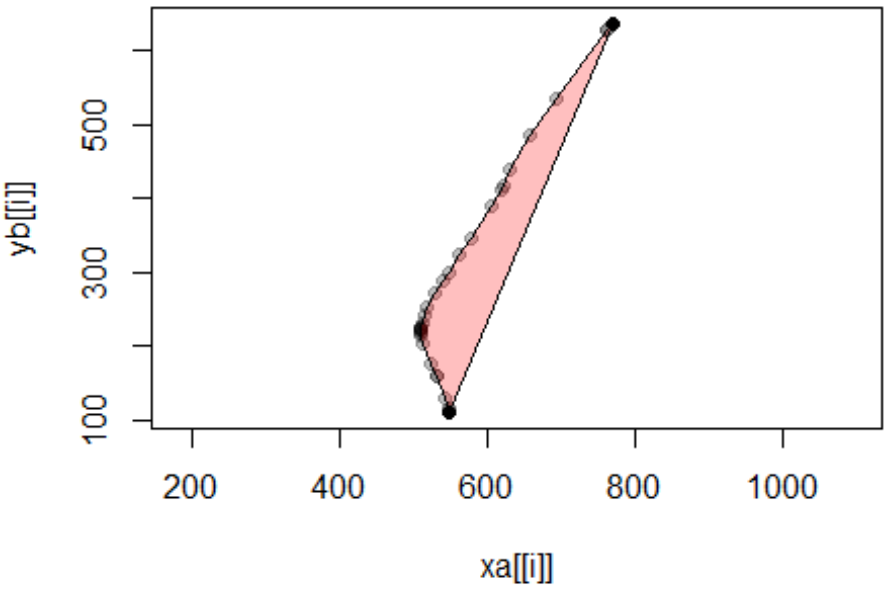

yb[[i]]
300
400
500
600
600
800
1000
1200
xa[[i]]

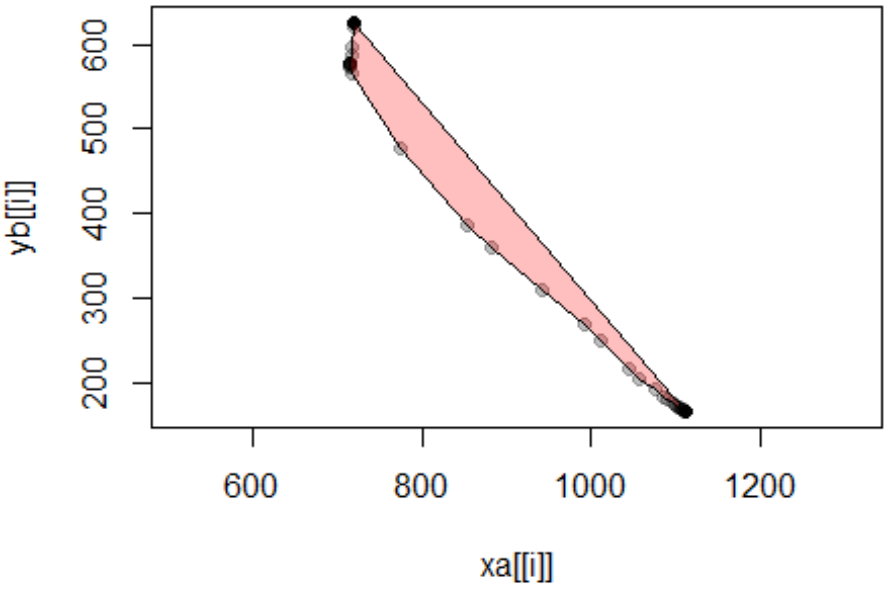

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

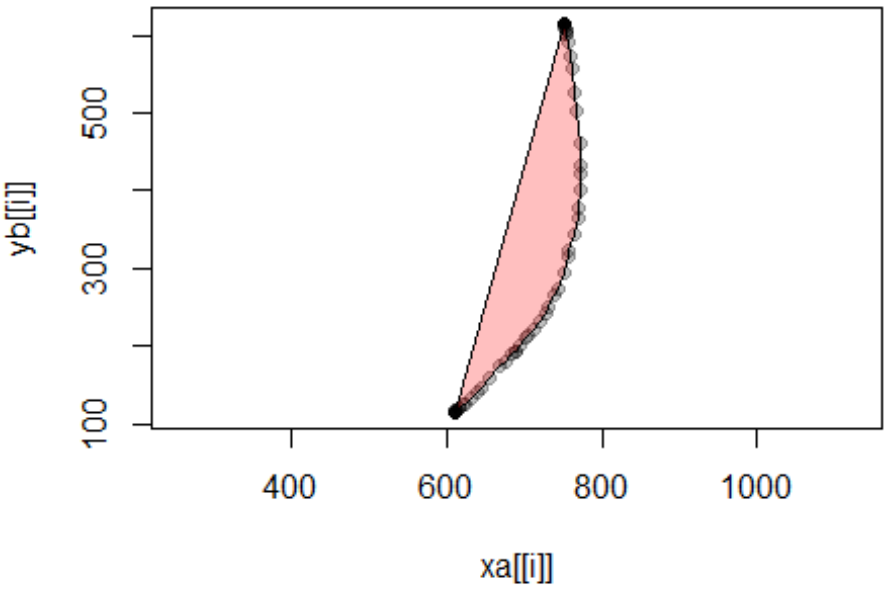

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

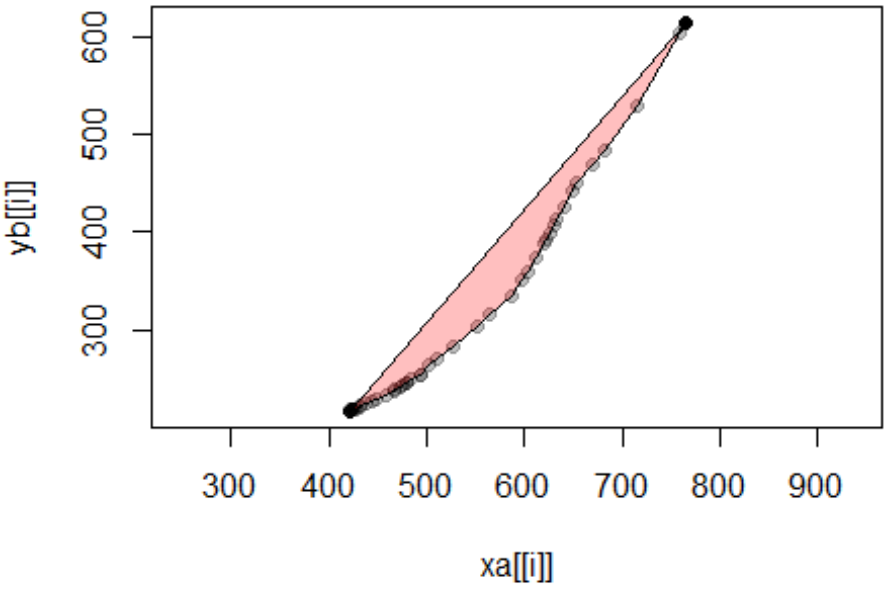

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

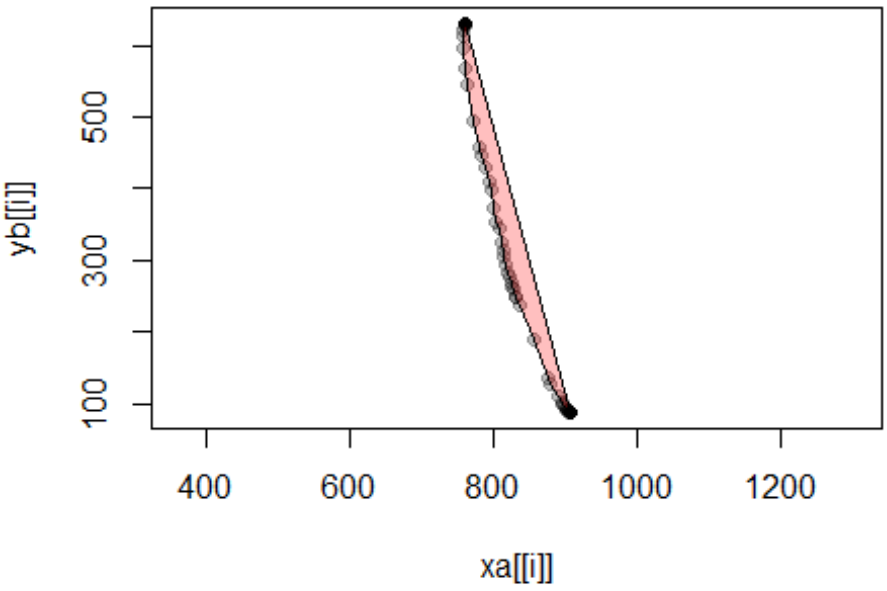

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
1300
xa[[i]]

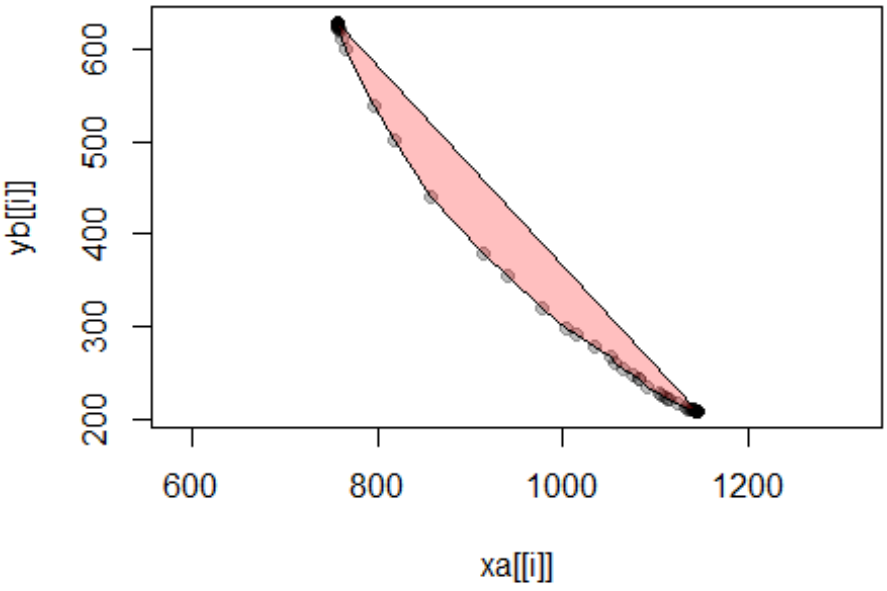

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

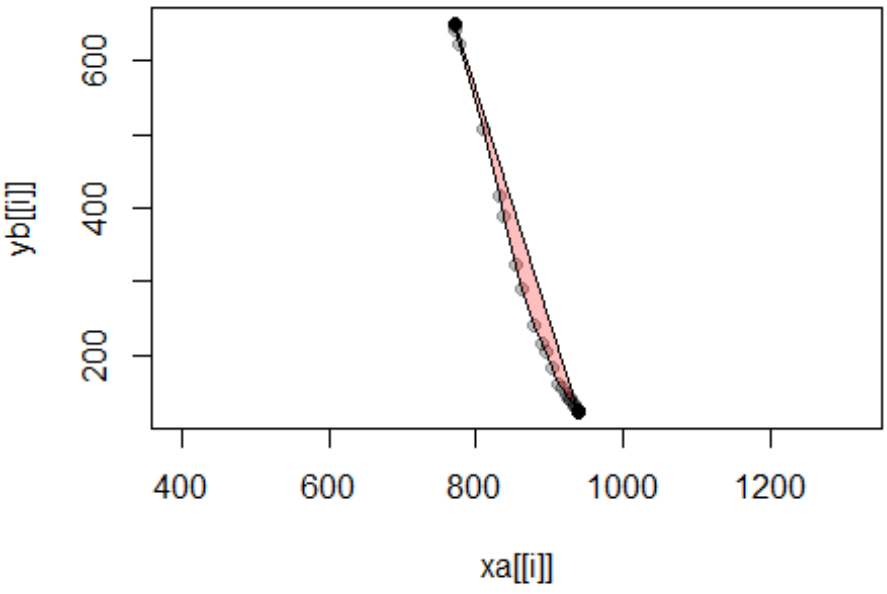

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

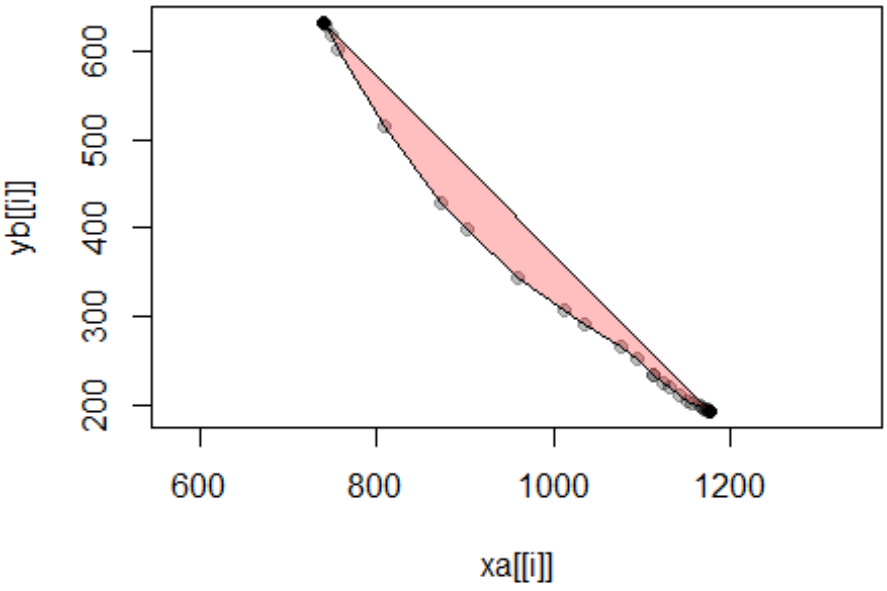

yb[[i]]
100
300
500
400
600
800
1000
xa[[i]]

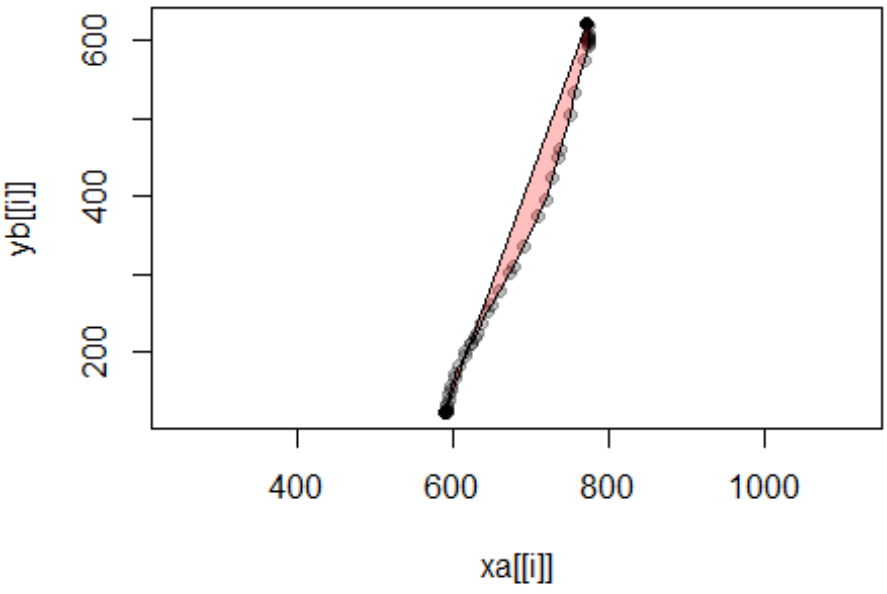

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

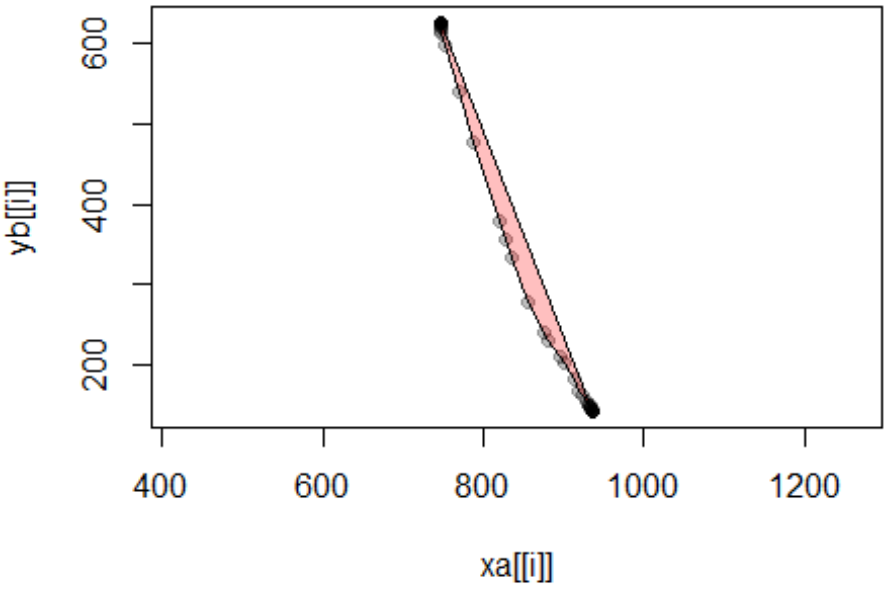

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

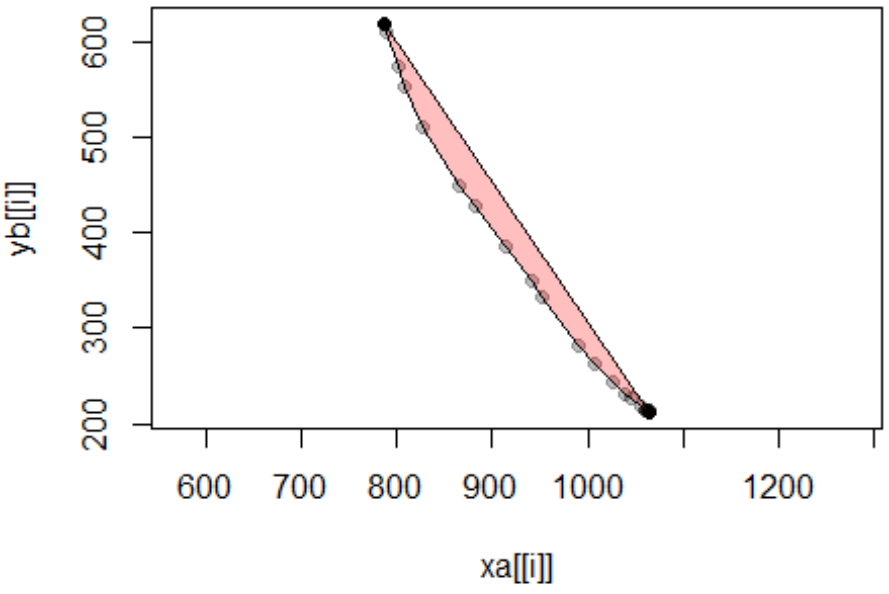

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

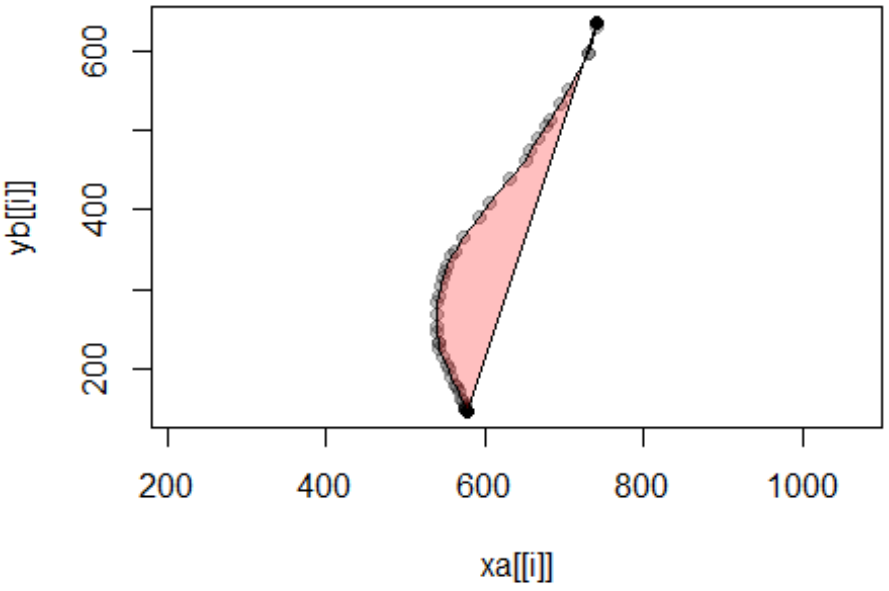

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

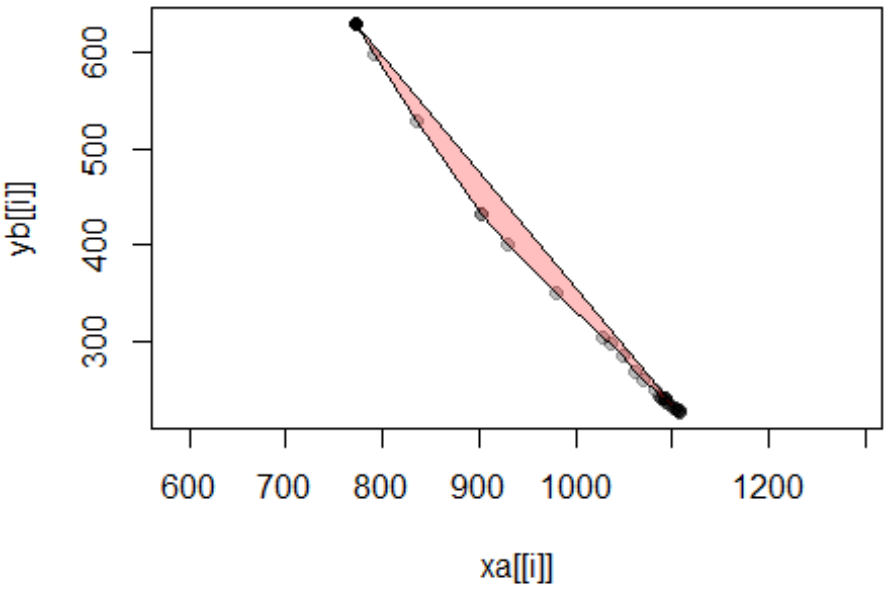

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

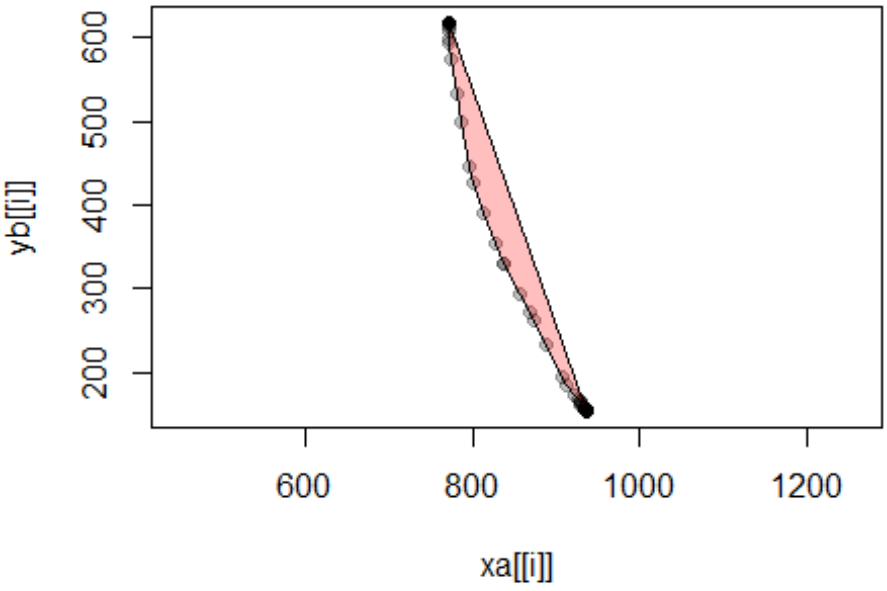

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

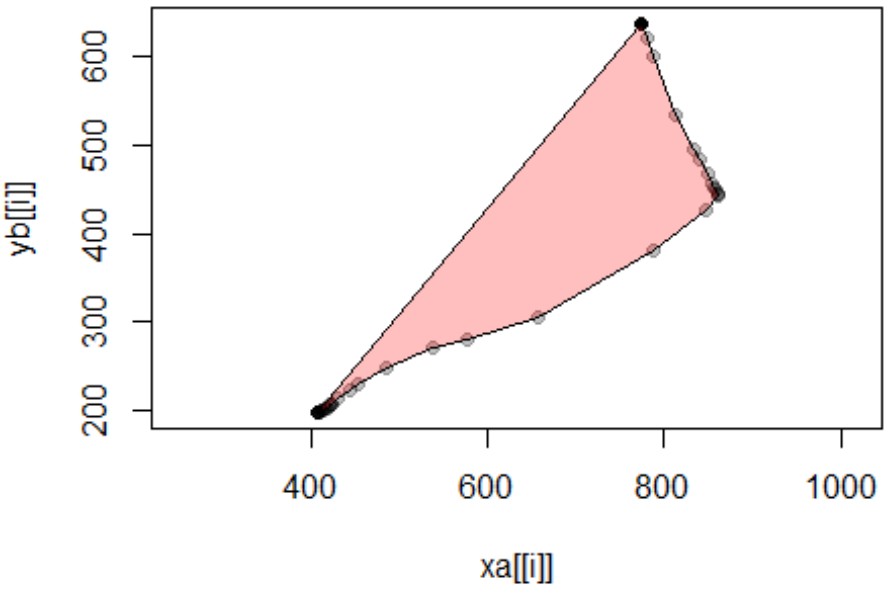

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

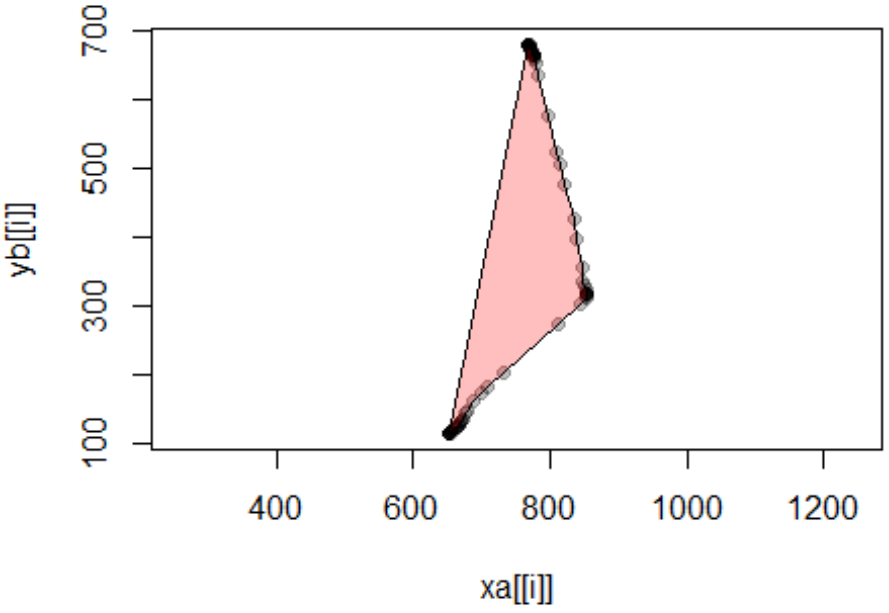

yb[[i]]
200
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

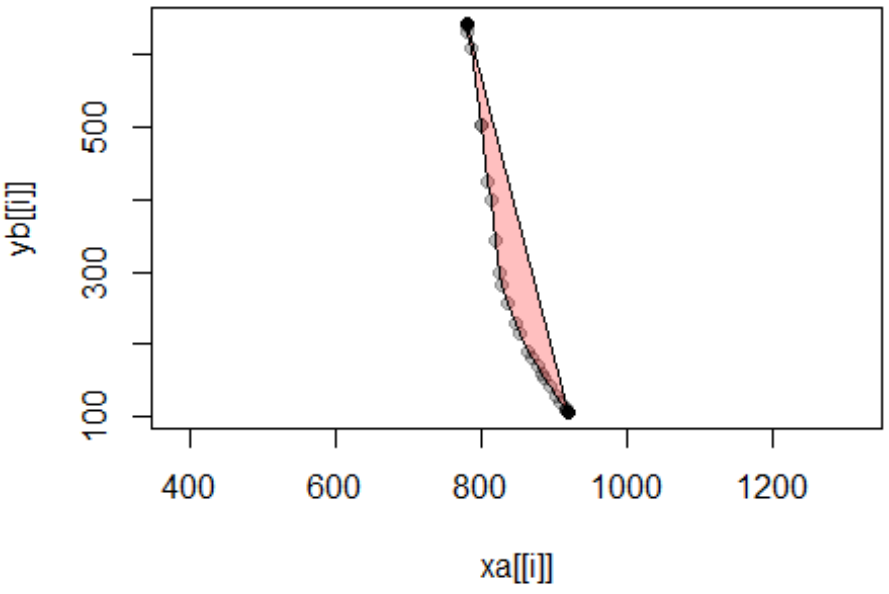

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

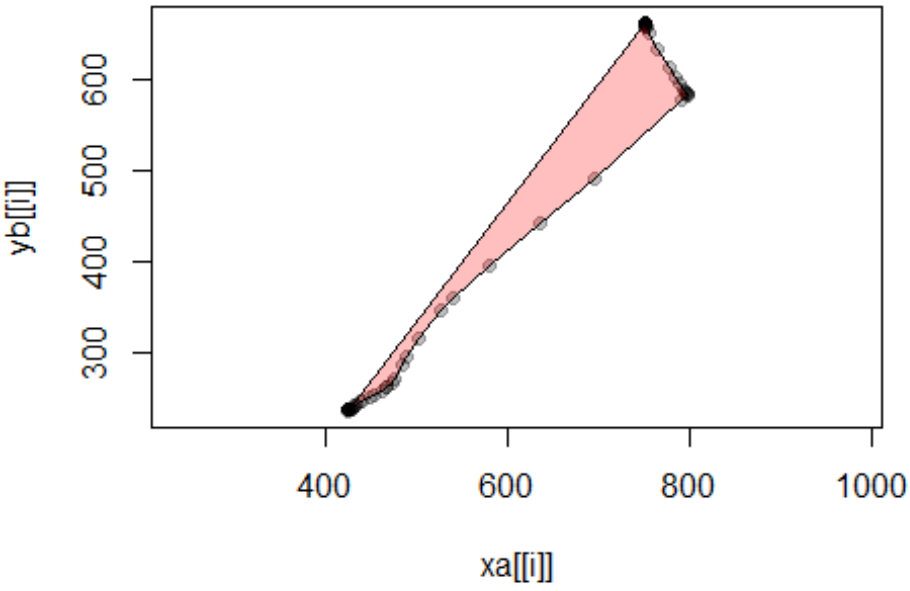

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

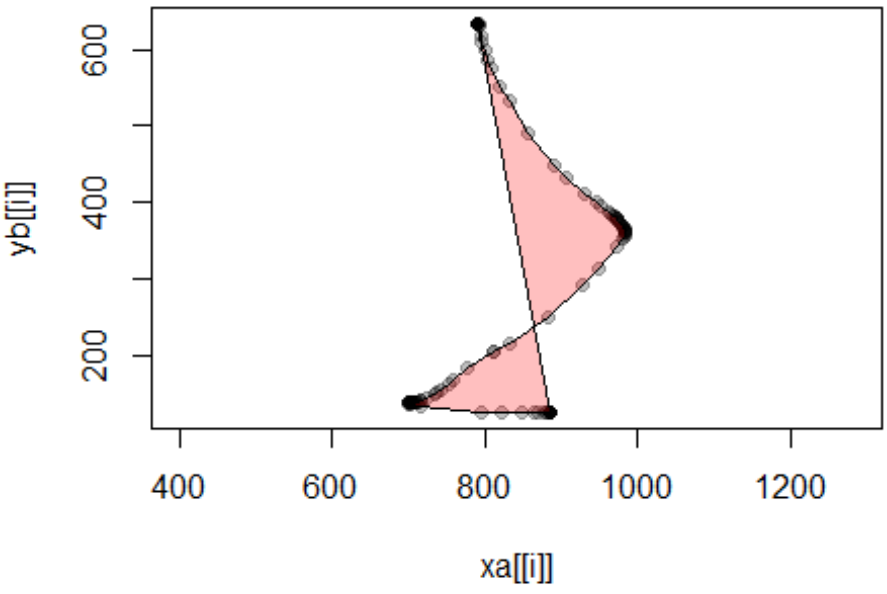

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

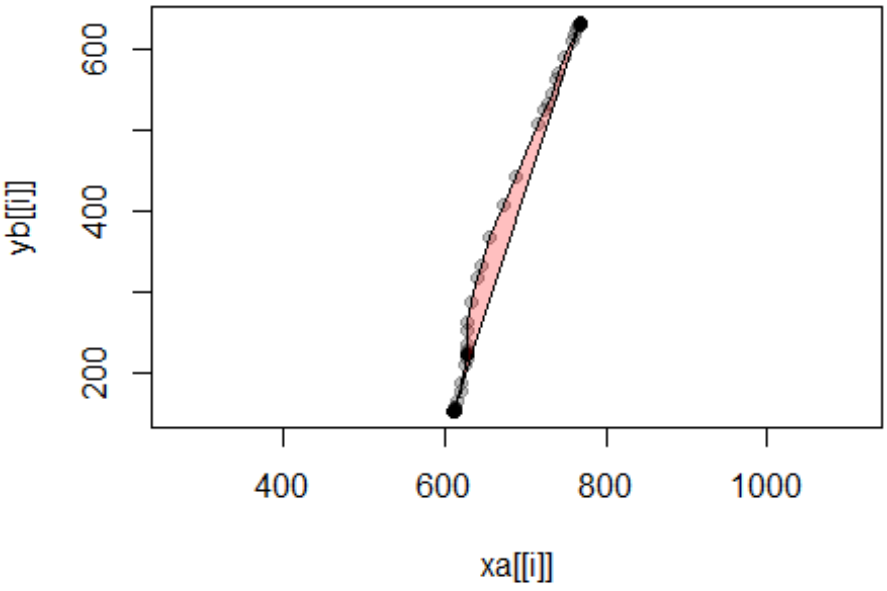

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

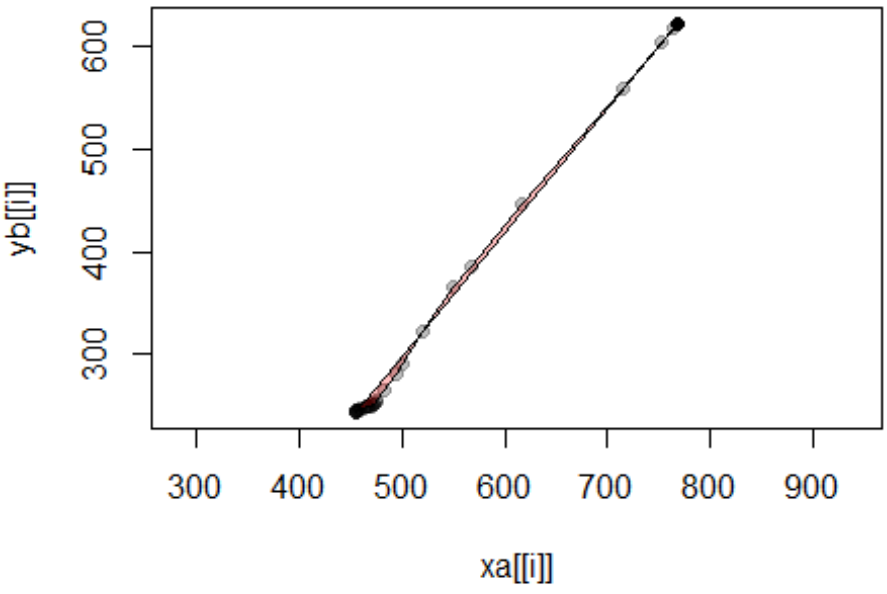

yb[[i]]
100
300
500
700
400
600
800
1000
1200
xa[[i]]

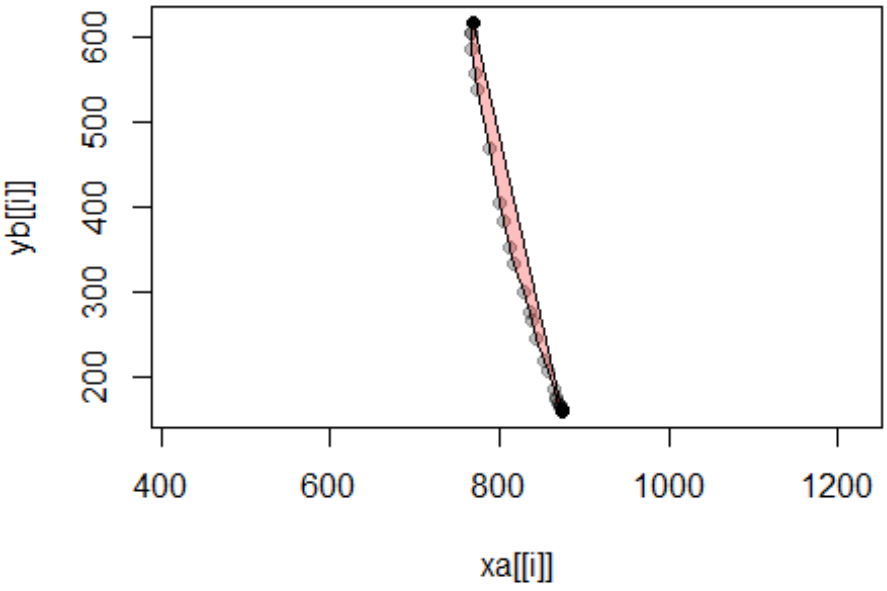

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

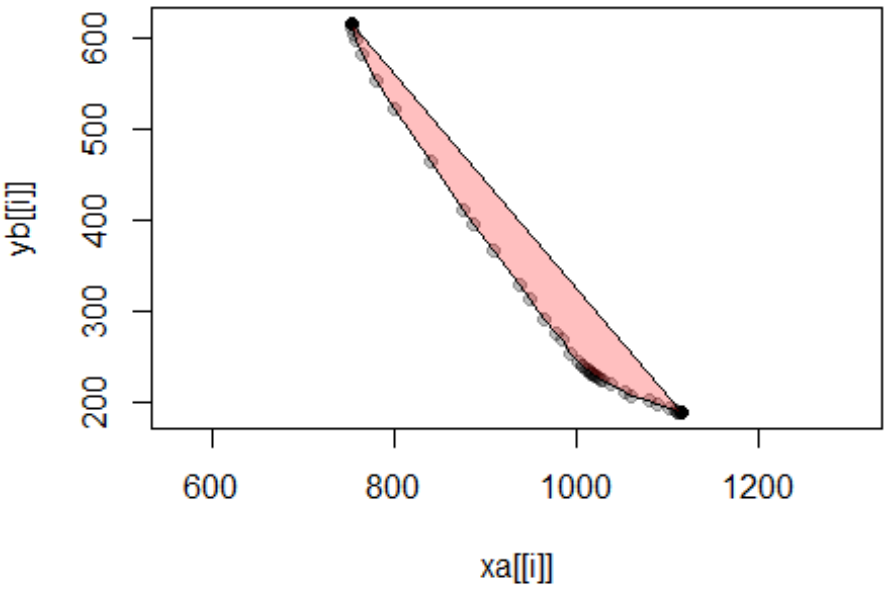

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

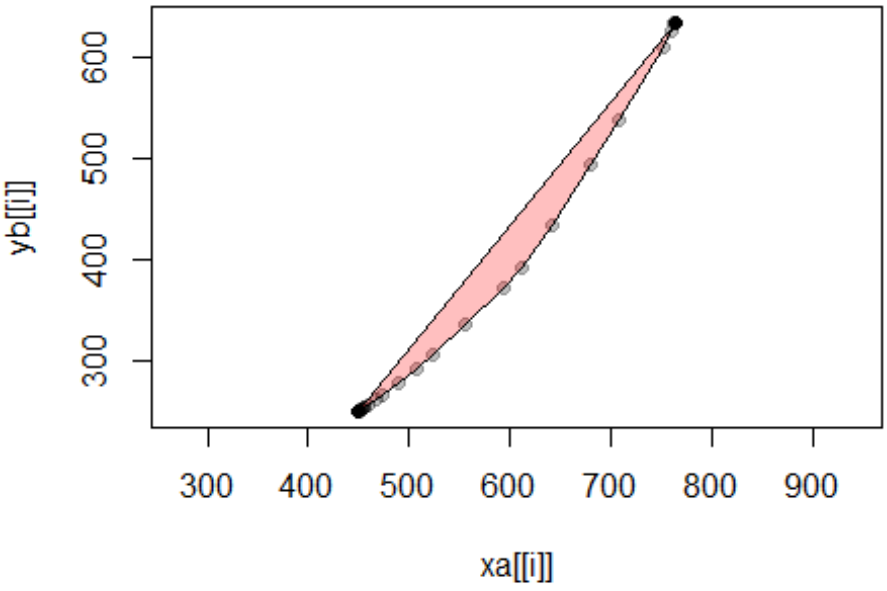

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

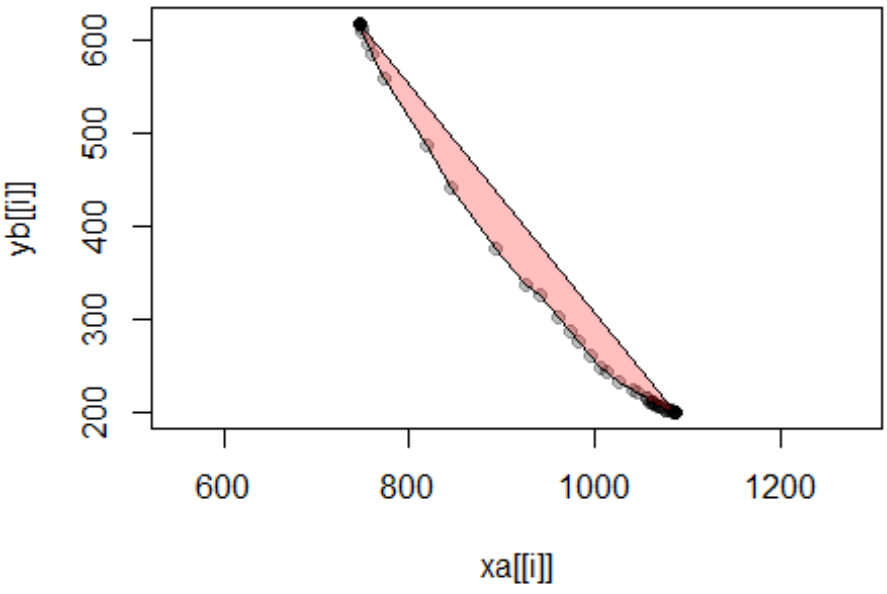

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

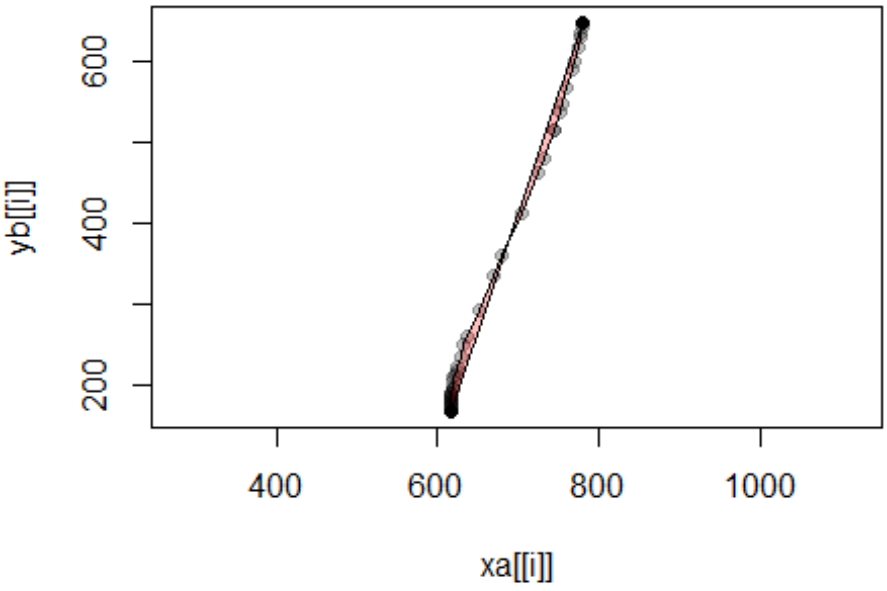

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

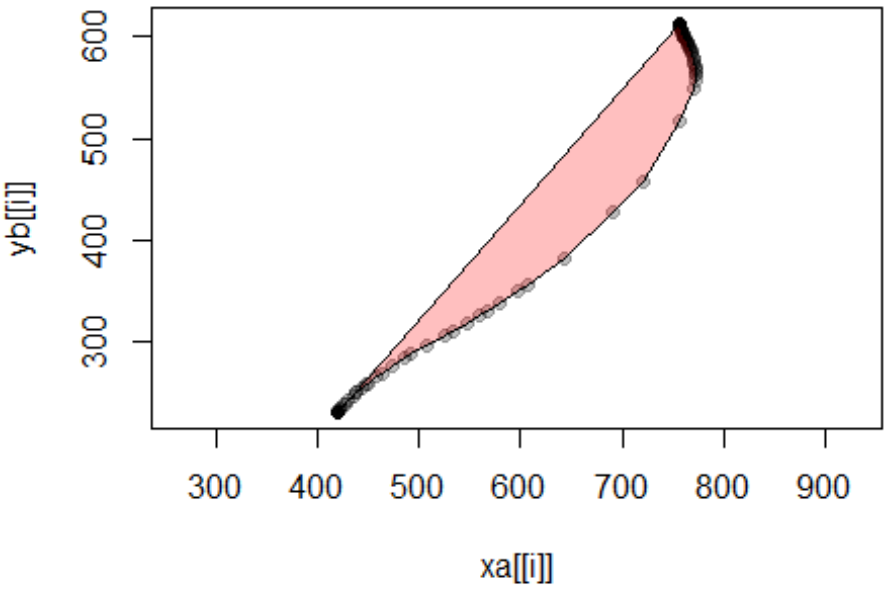

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

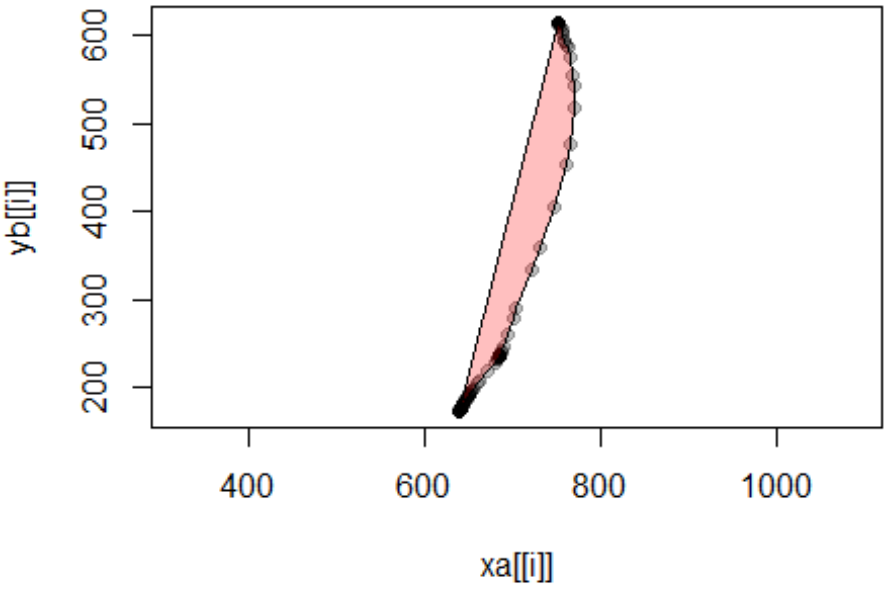

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

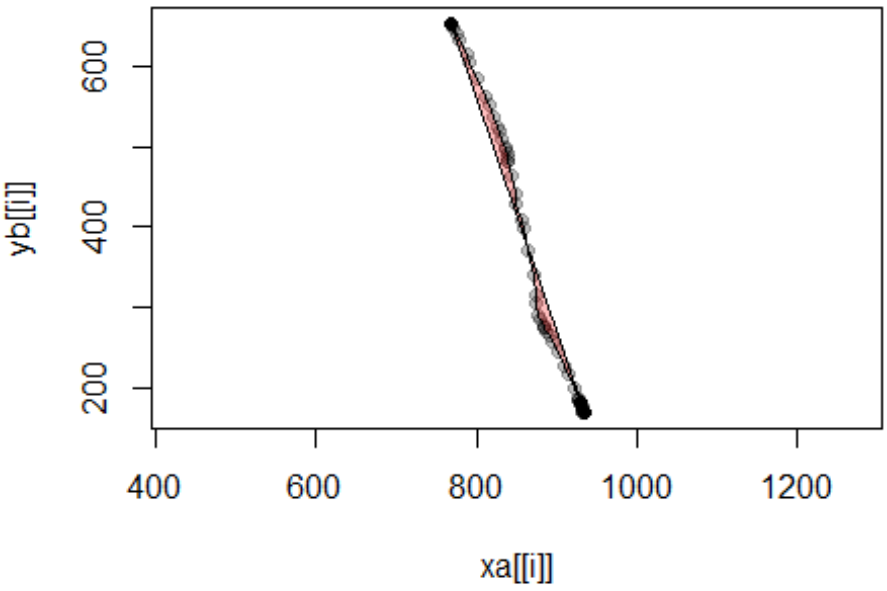

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

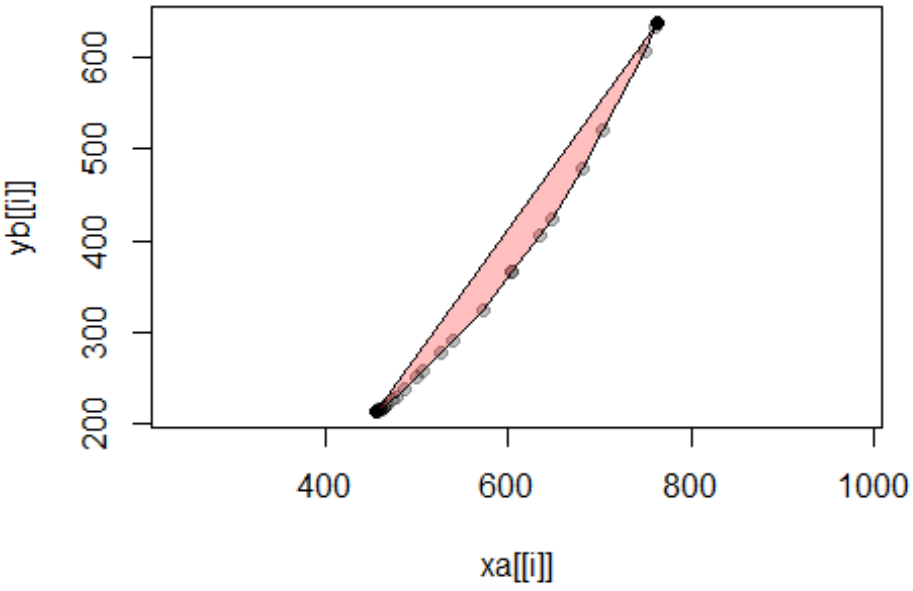

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

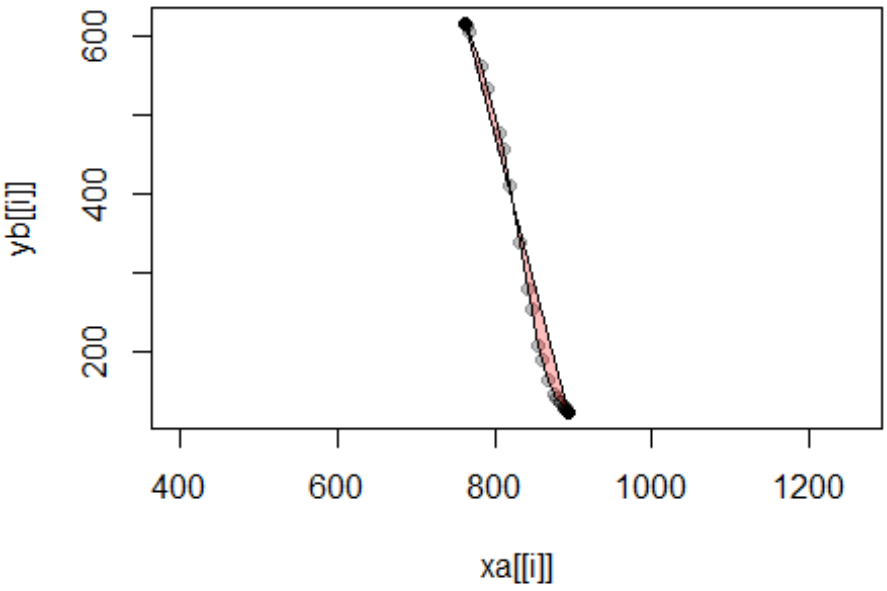

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

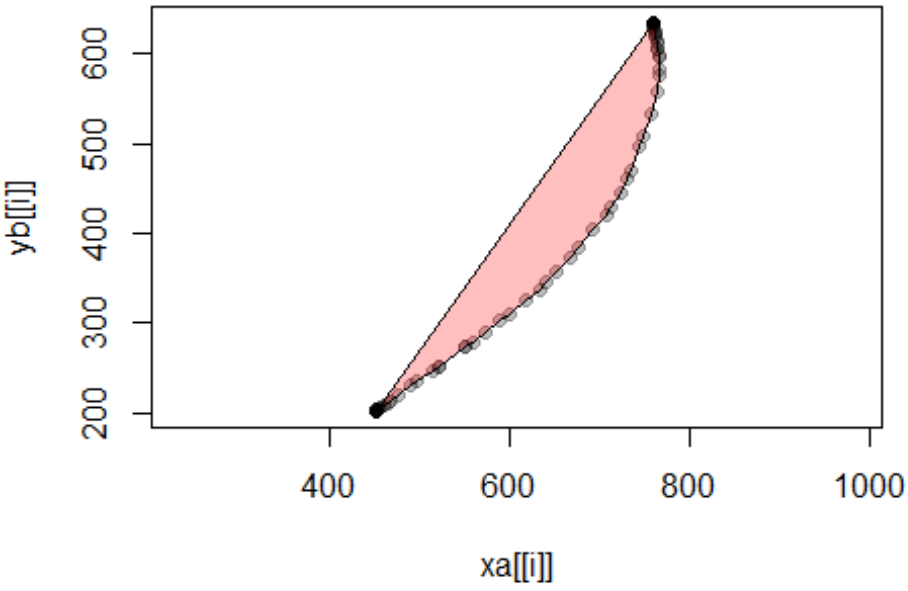

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

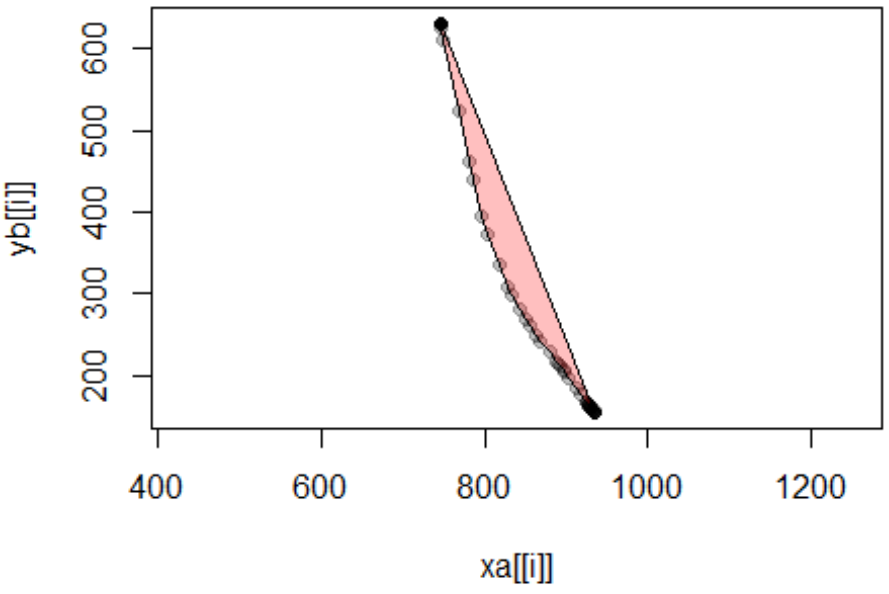

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

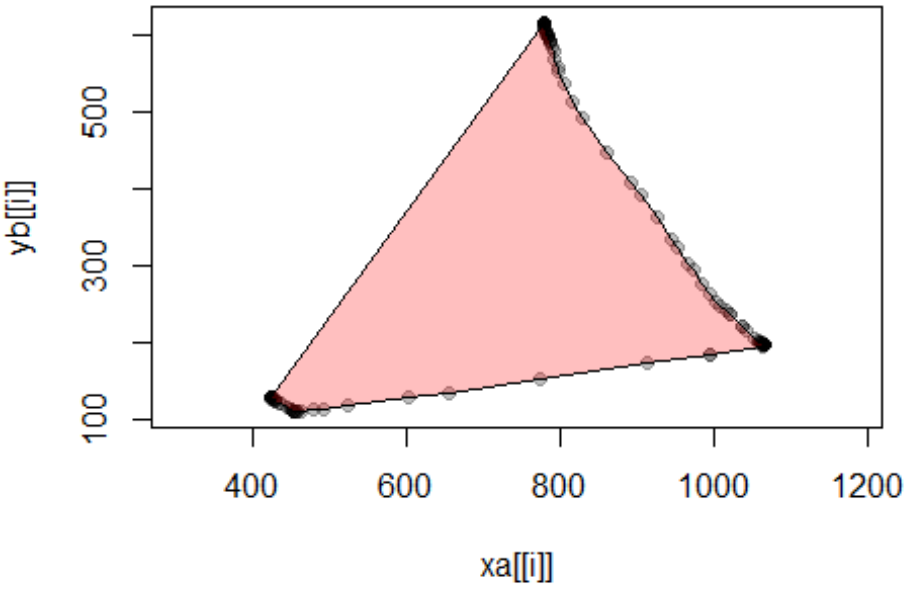

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

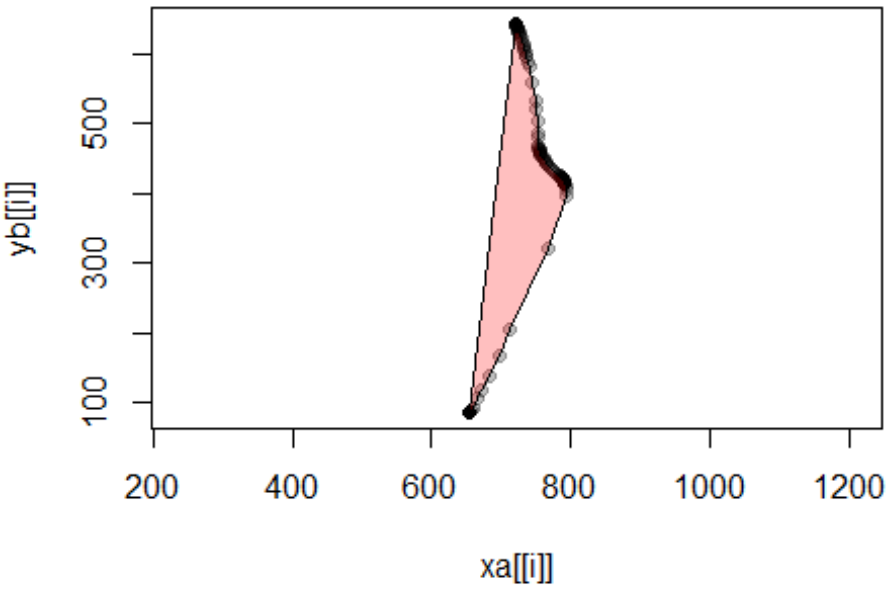

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

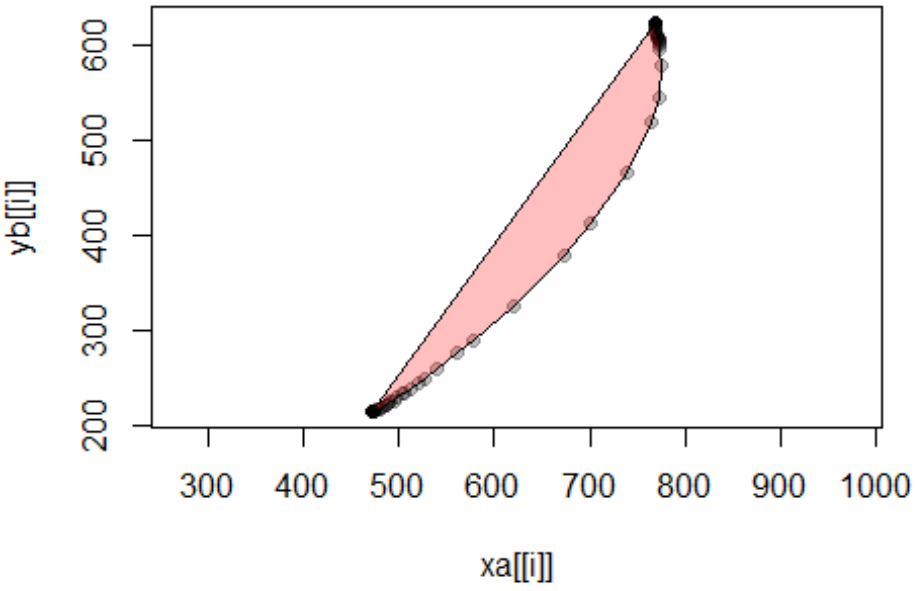

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

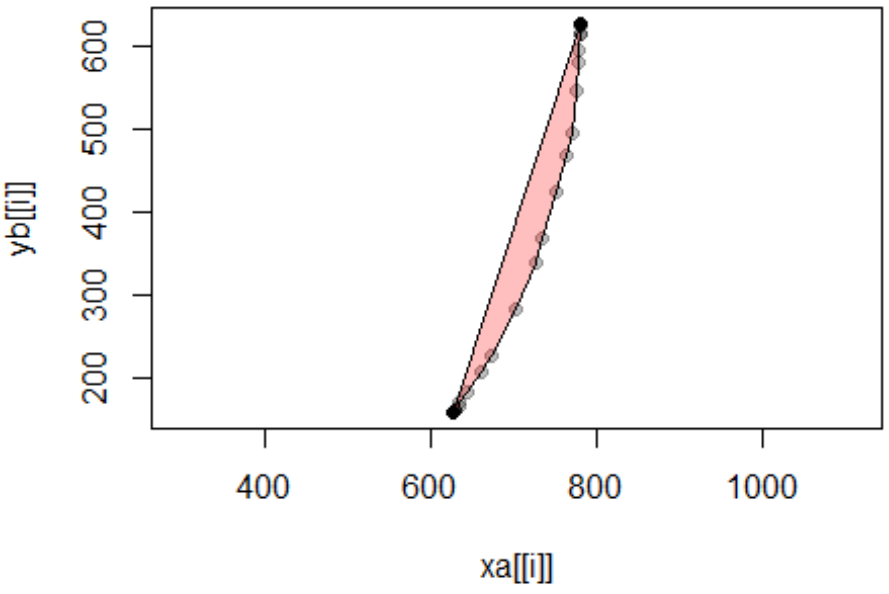

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

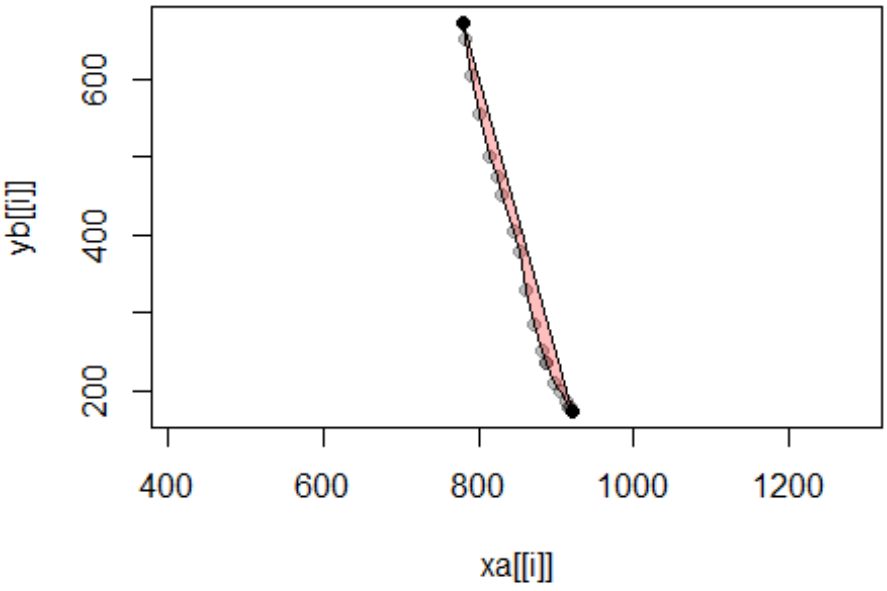

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

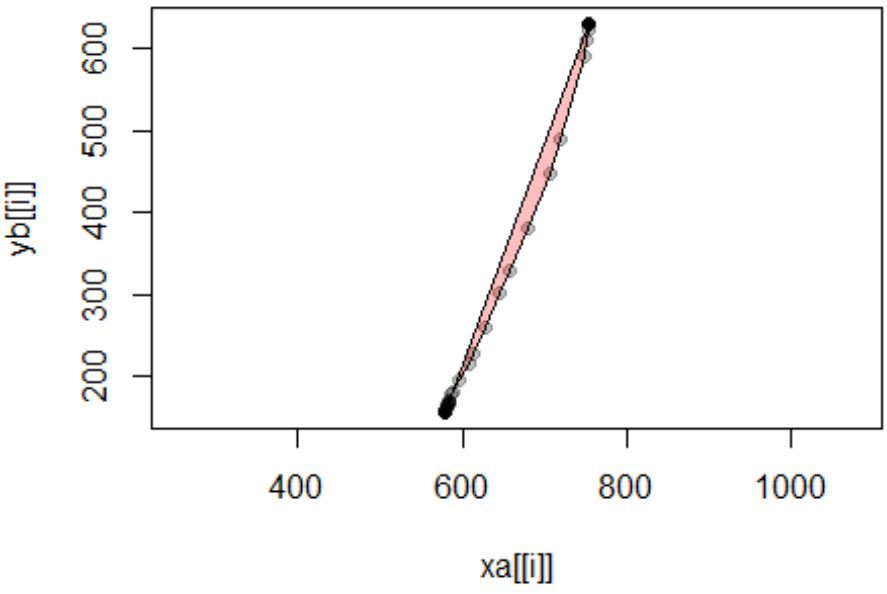

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

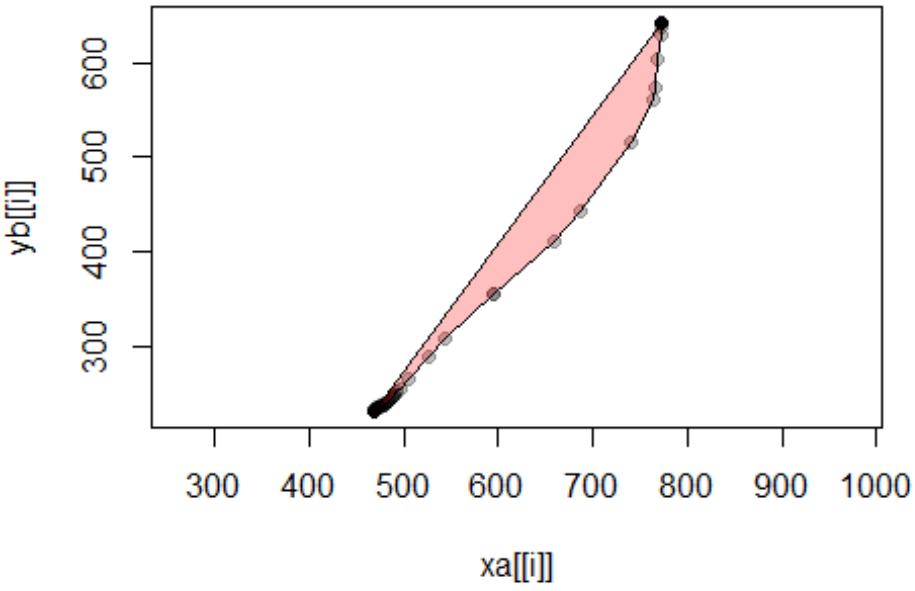

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

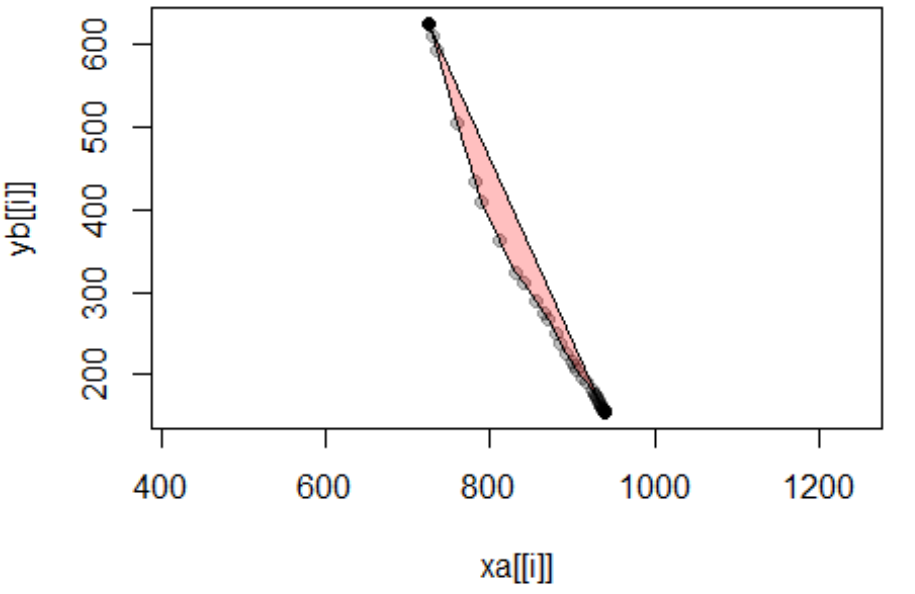

yb[[i]]
200
300
400
500
600
300
400
500
600
700
800
900
1000
xa[[i]]

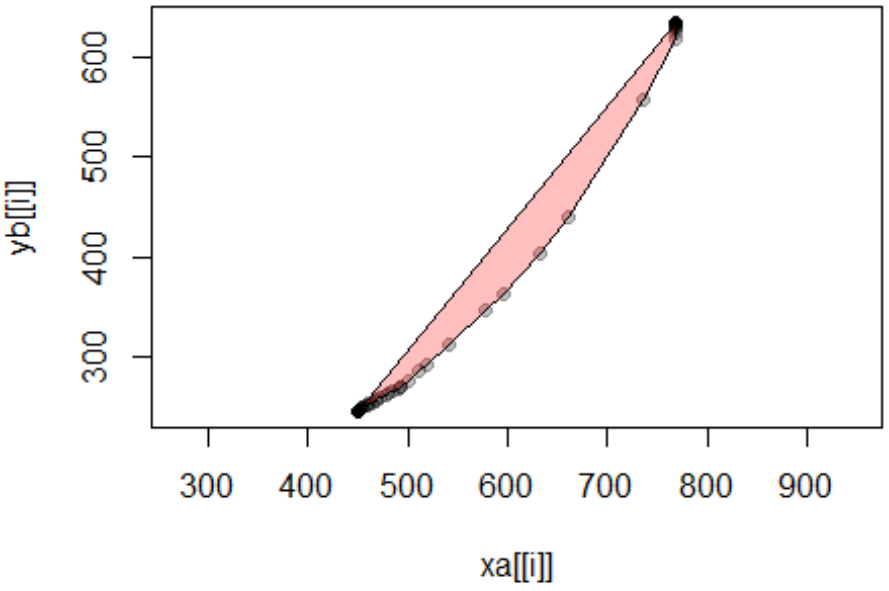

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

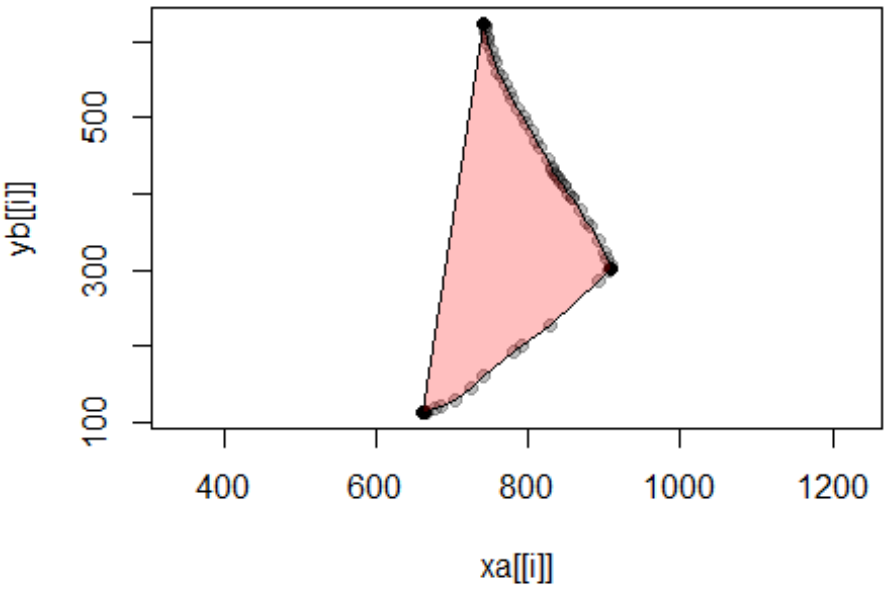

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

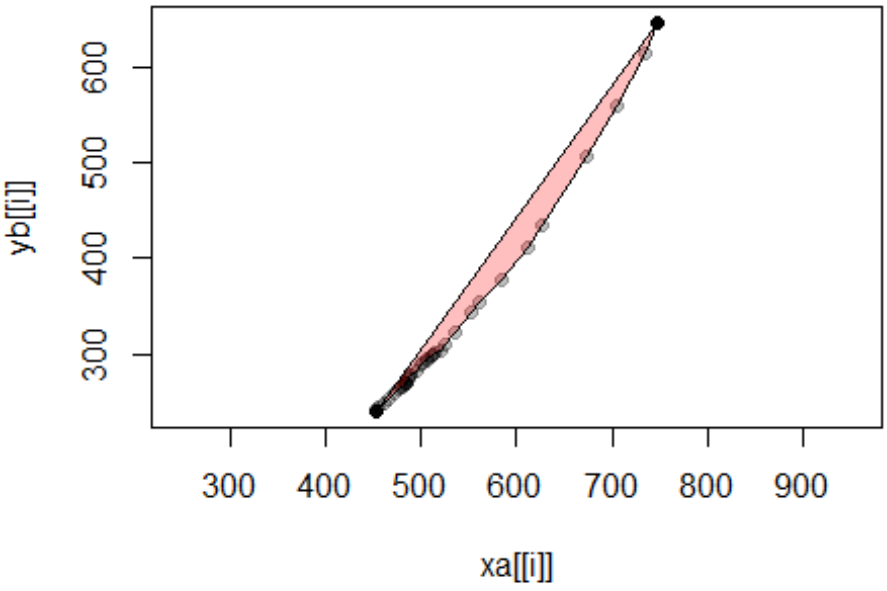

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

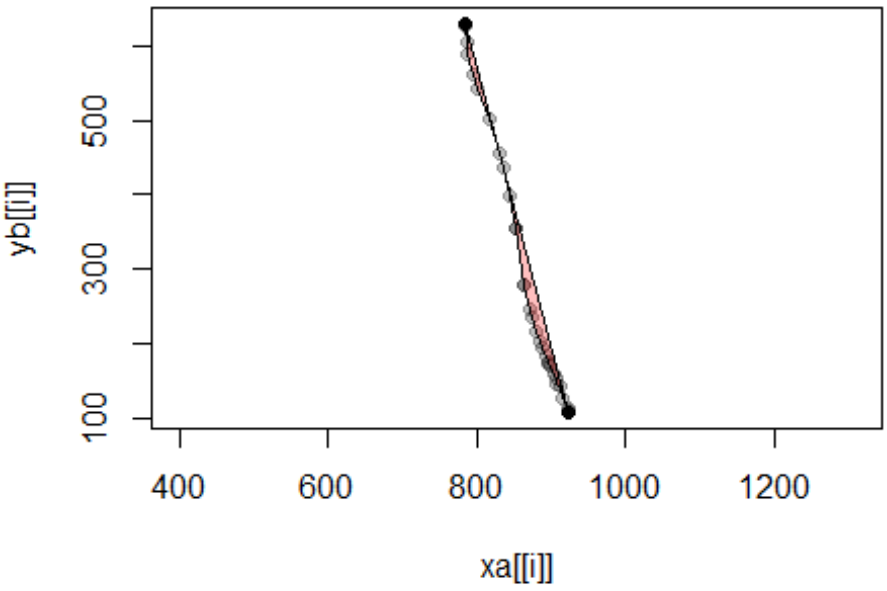

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
1000
xa[[i]]

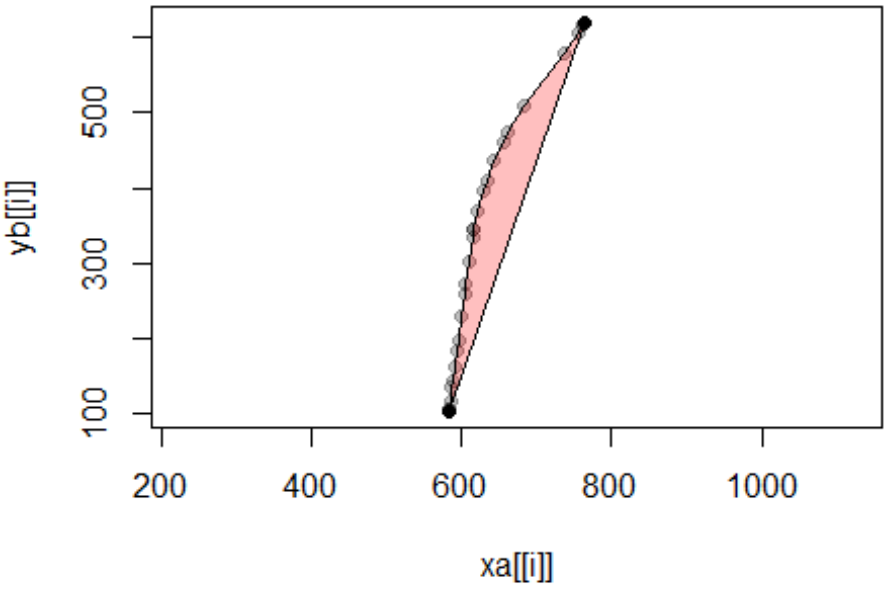

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

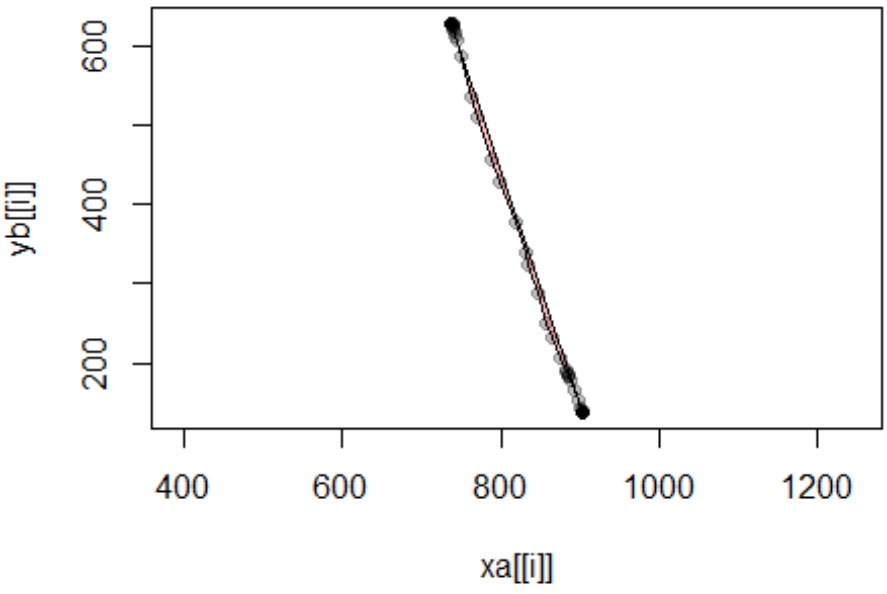

yb[[i]]
300
400
500
600
xa[[i]]
300
400
500
600
700
800
900

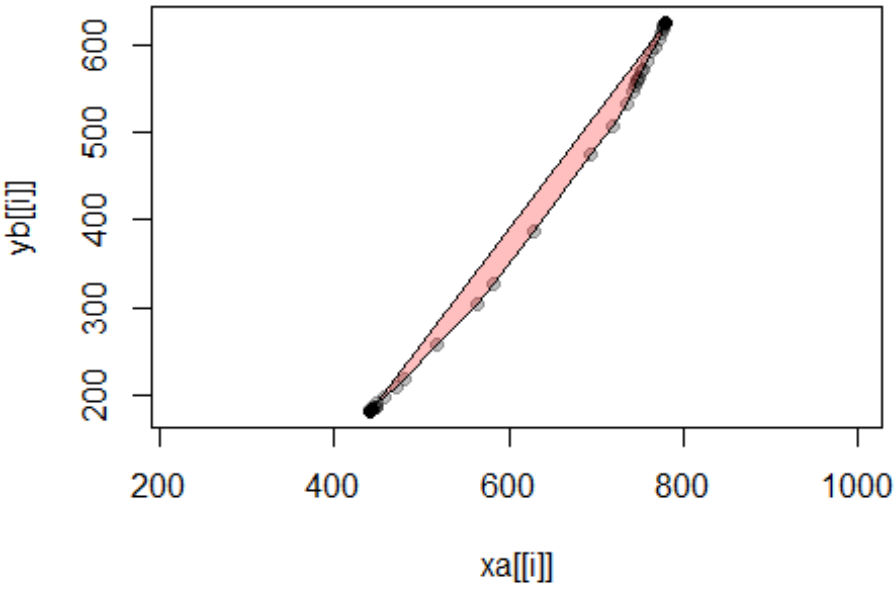

yb[[i]]
100
300
500
xa[[i]]
400
600
800
1000
1200

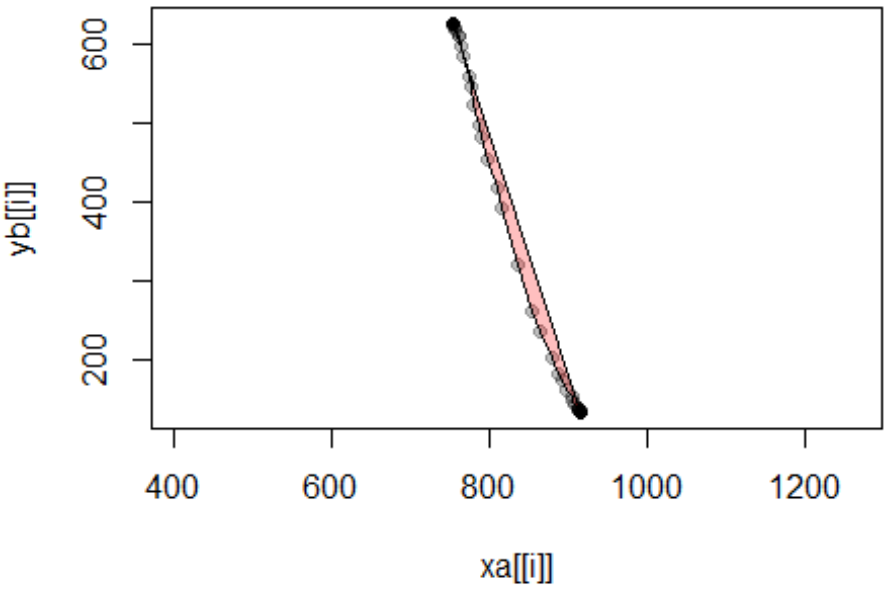

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
xa[[i]]

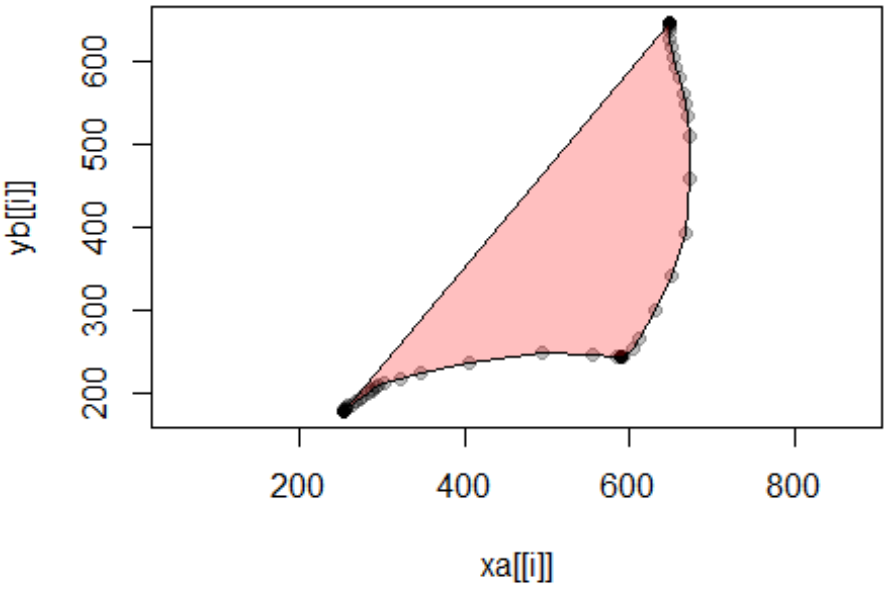

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

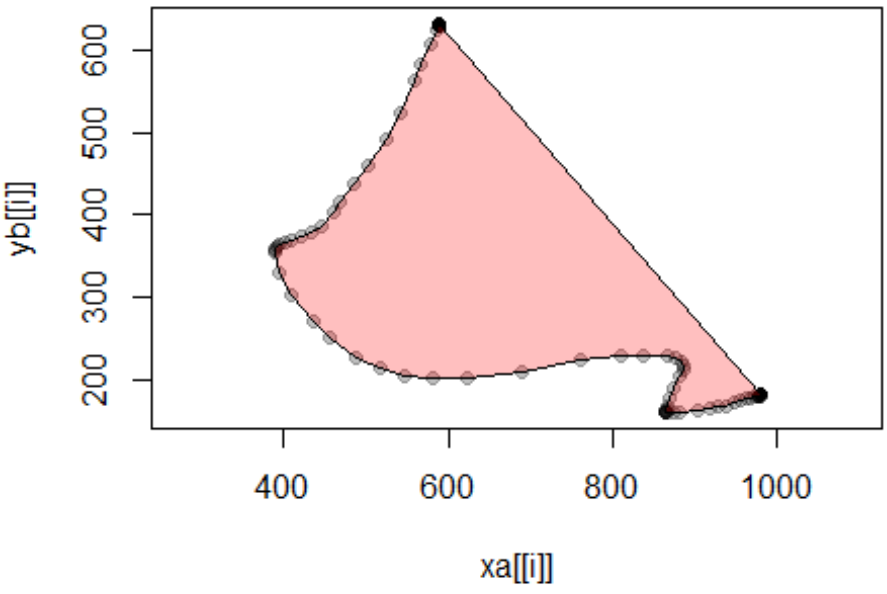

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

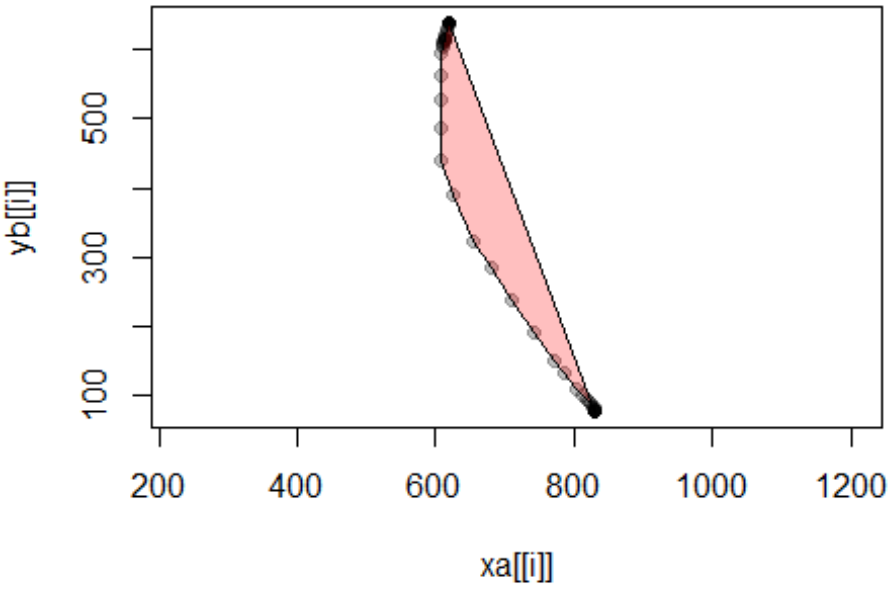

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

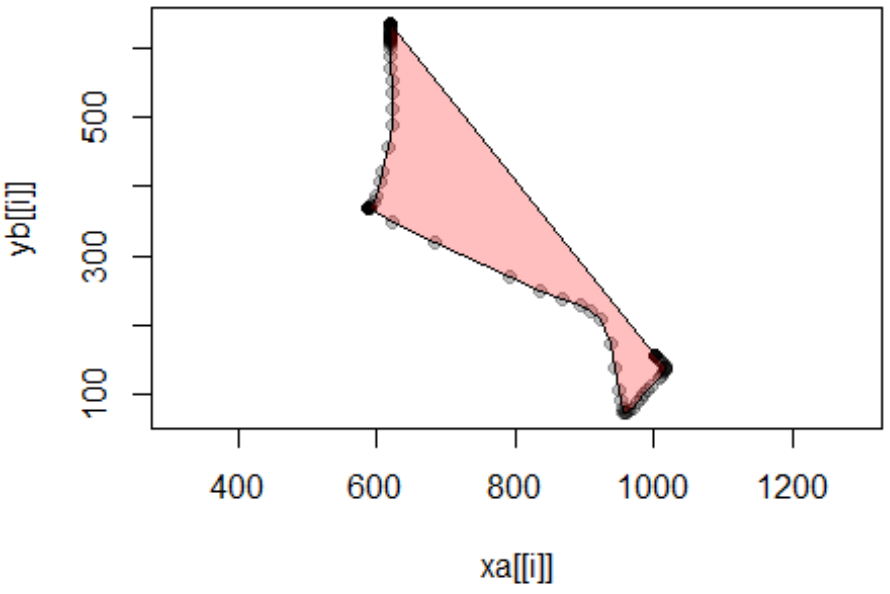

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

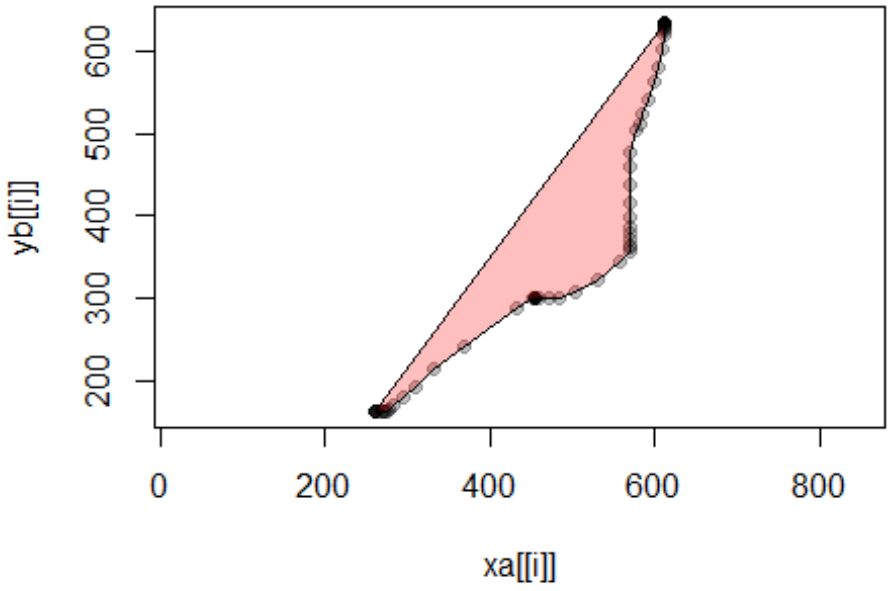

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

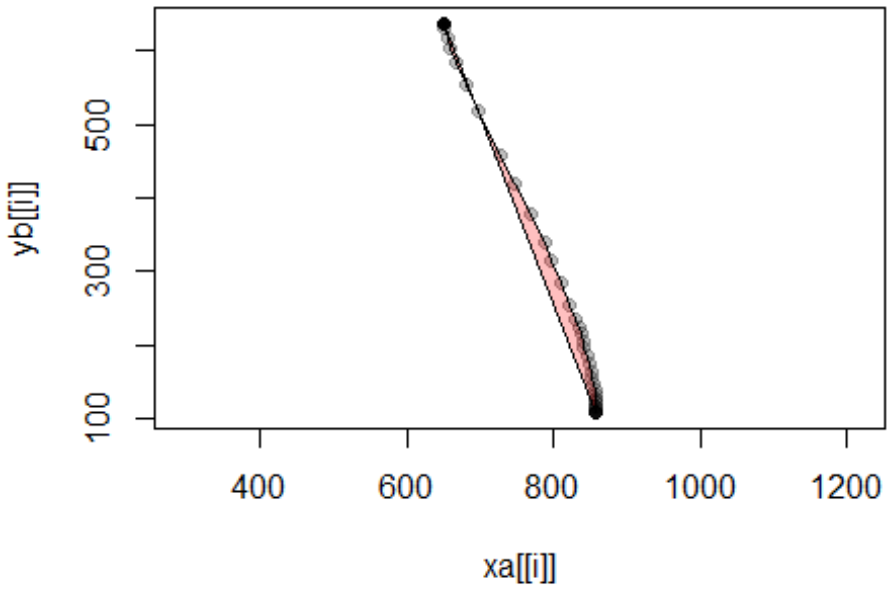

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

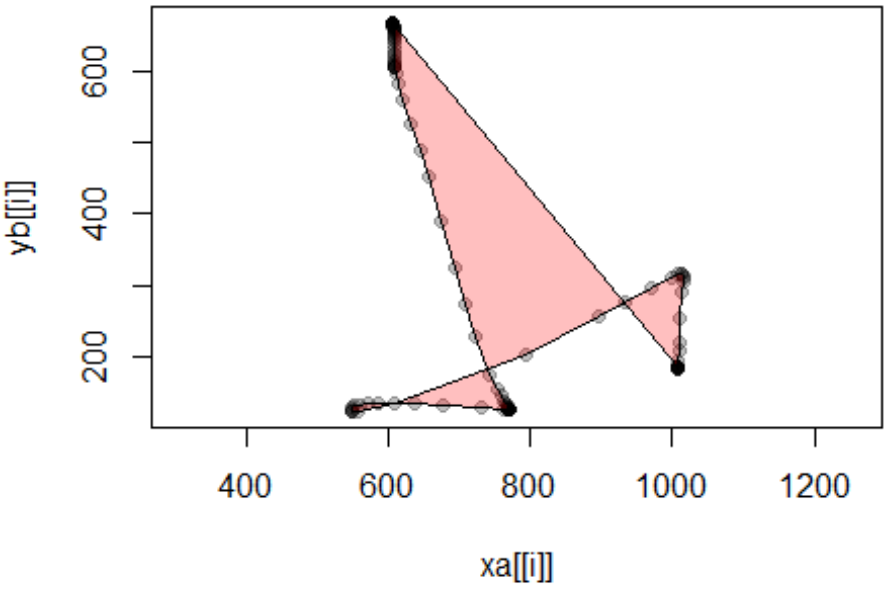

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

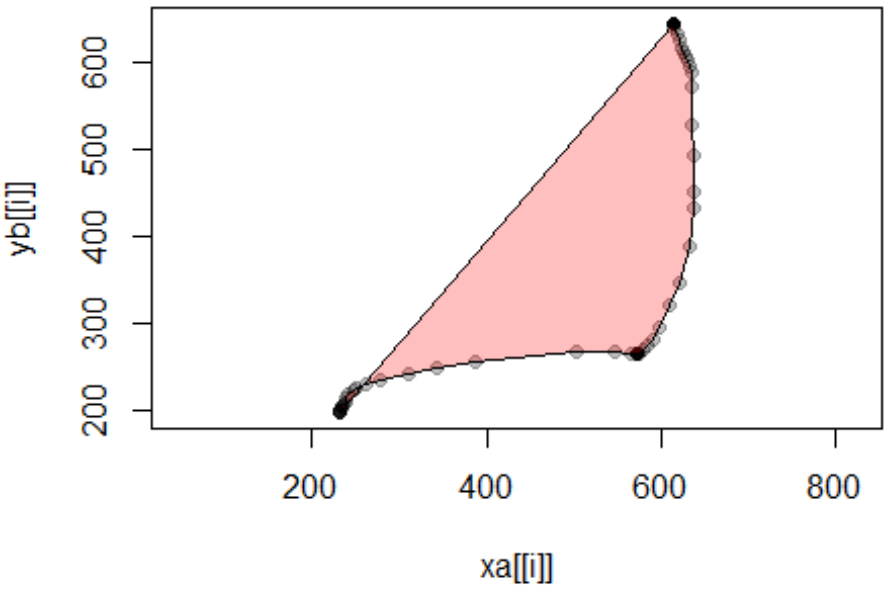

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

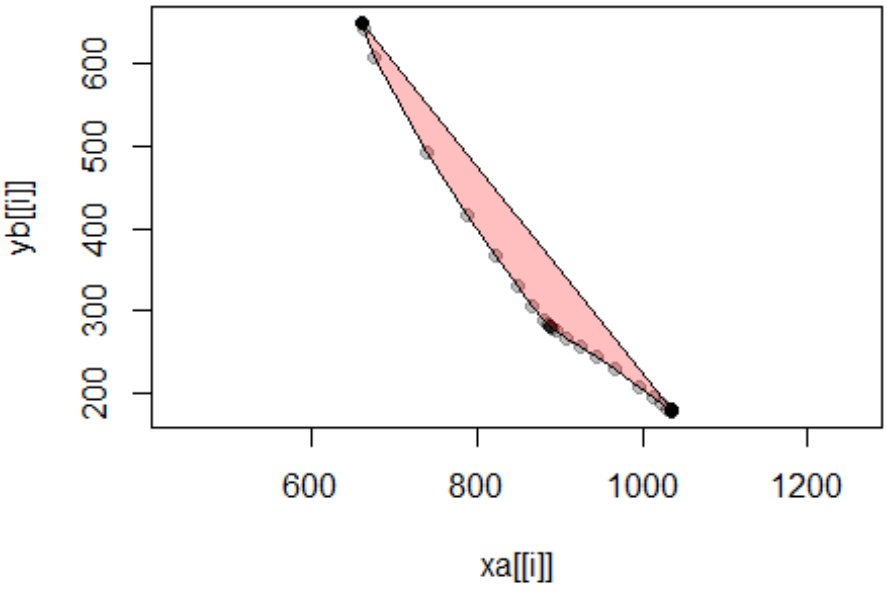

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

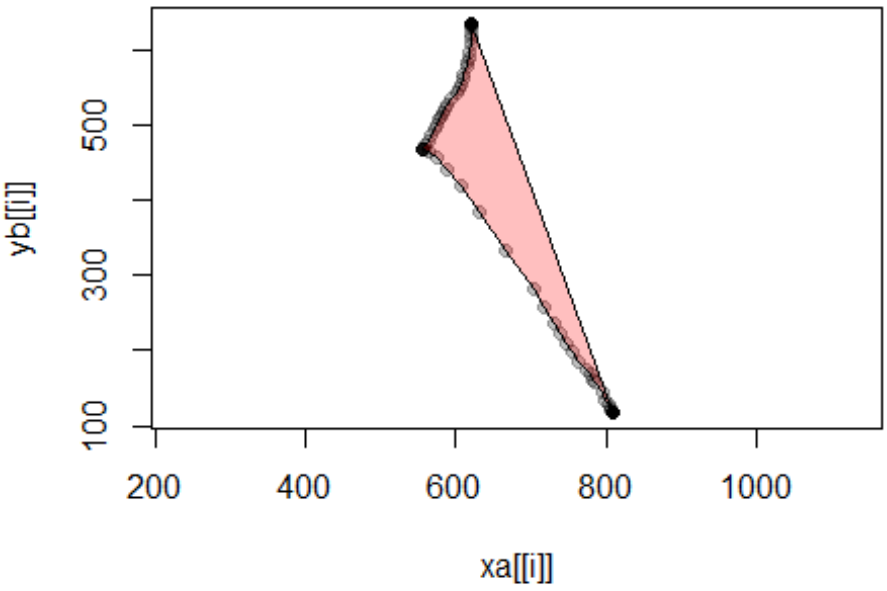

yb[[i]]
200
300
400
500
600
0
200
400
600
800
xa[[i]]

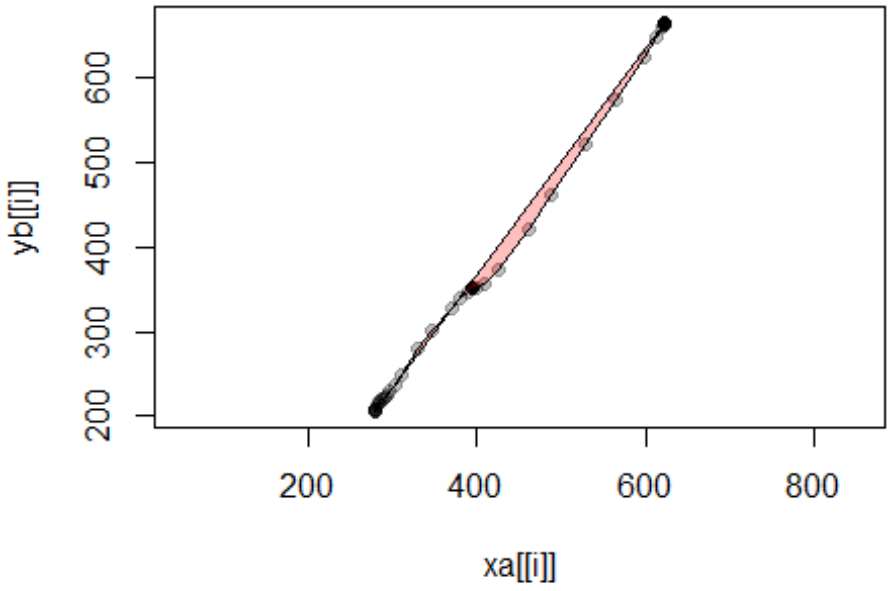

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

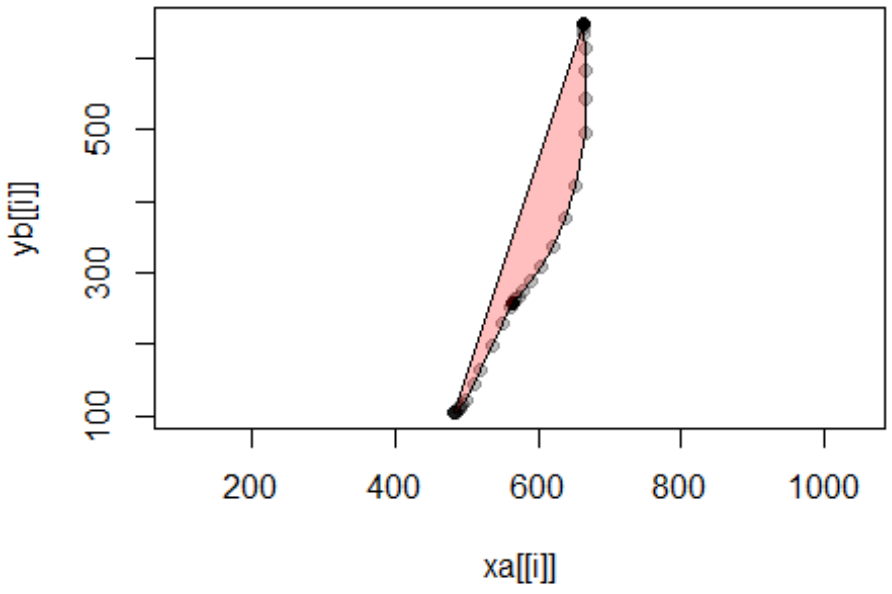

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

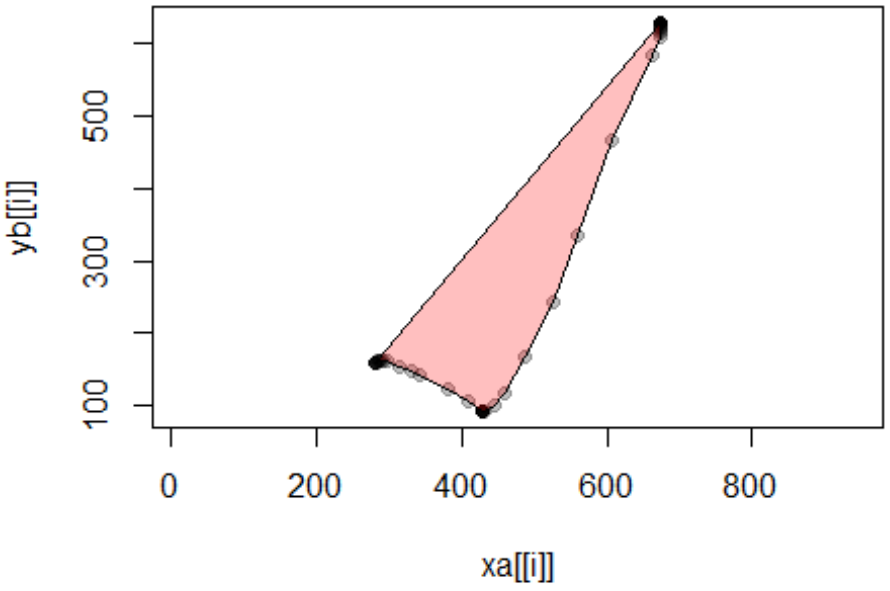

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

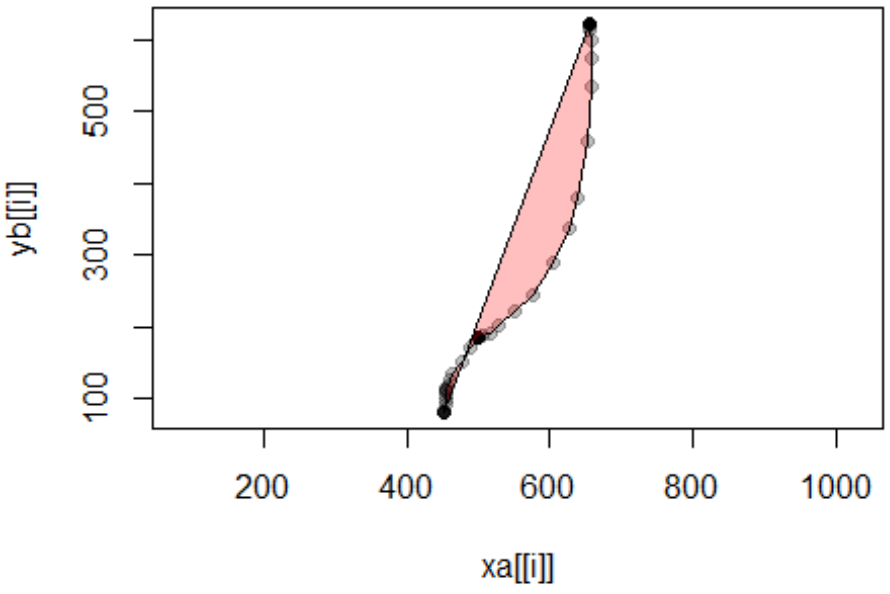

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

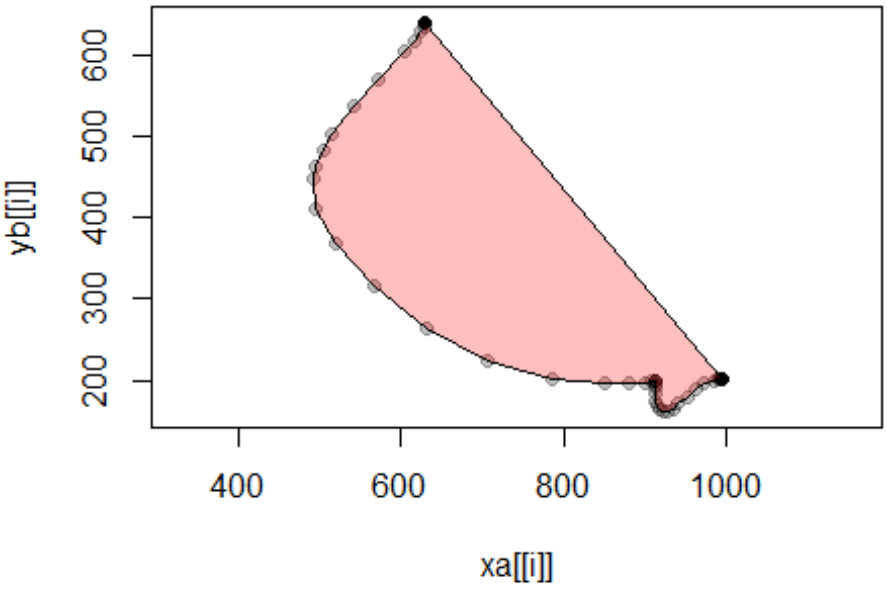

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

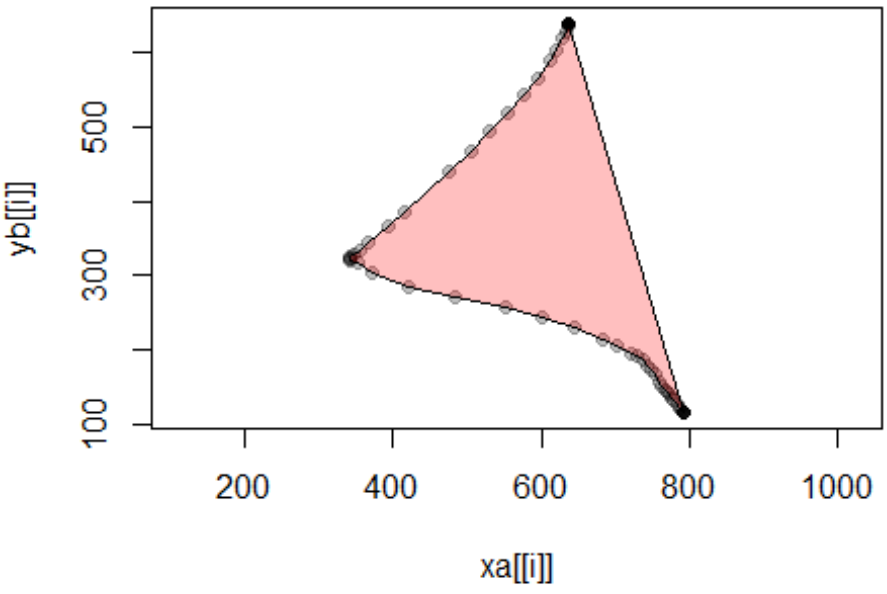

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

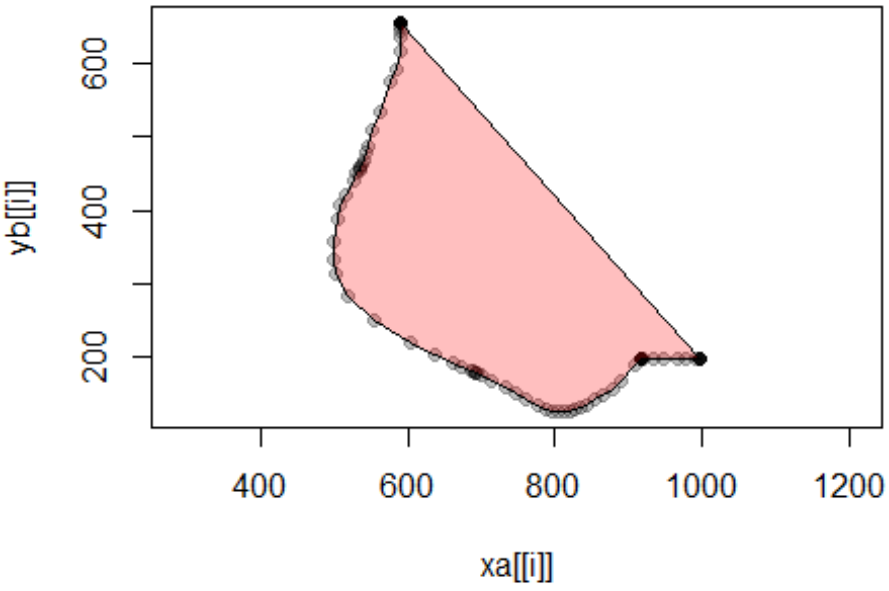

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

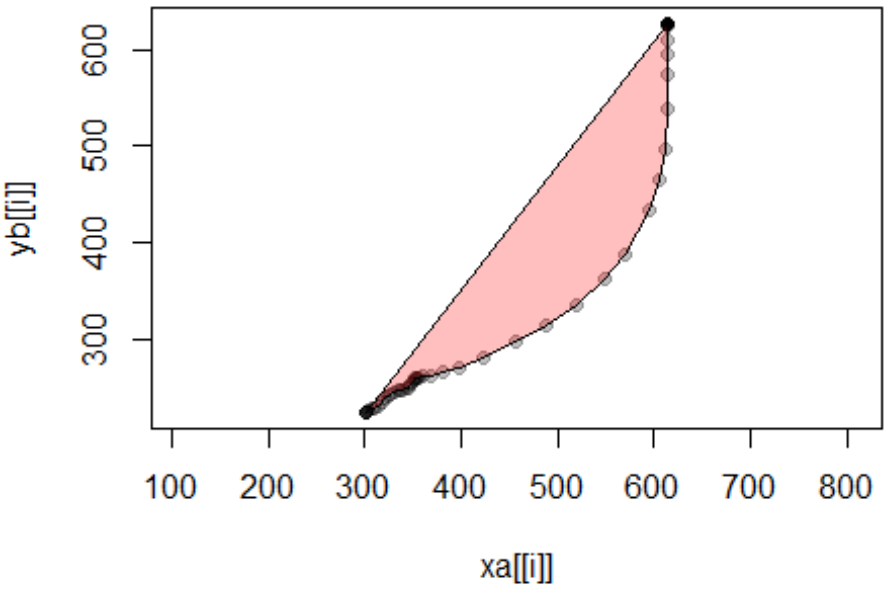

yb[[i]]
100
300
500
0
200
400
600
800
xa[[i]]

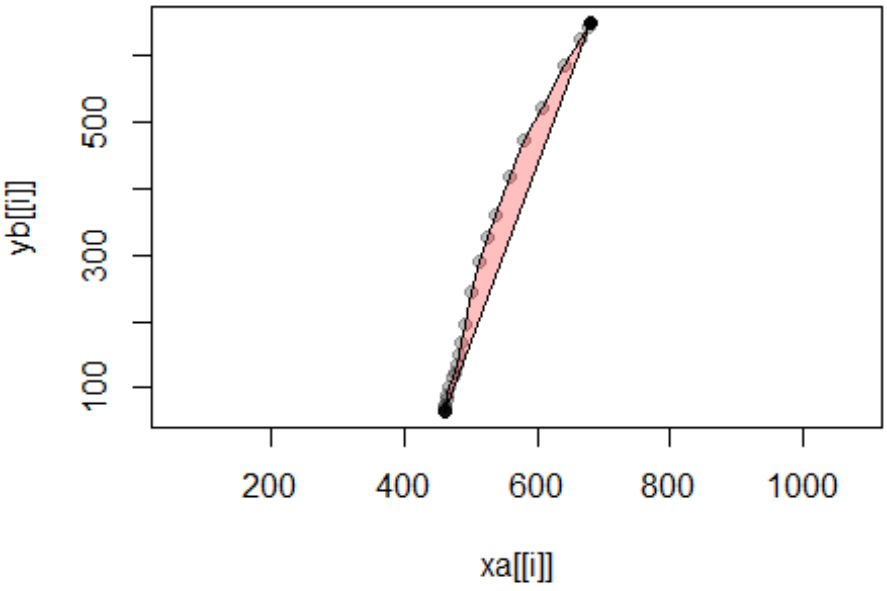

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

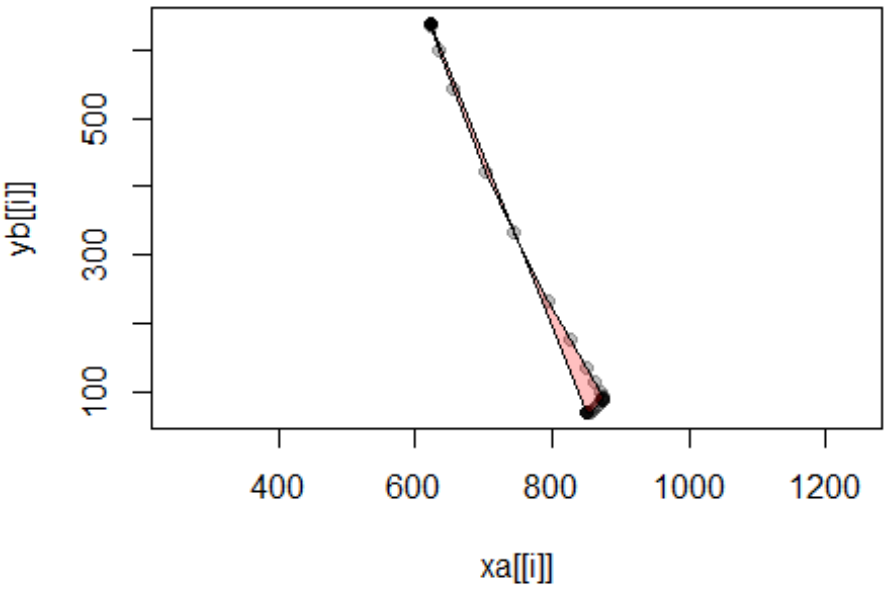

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

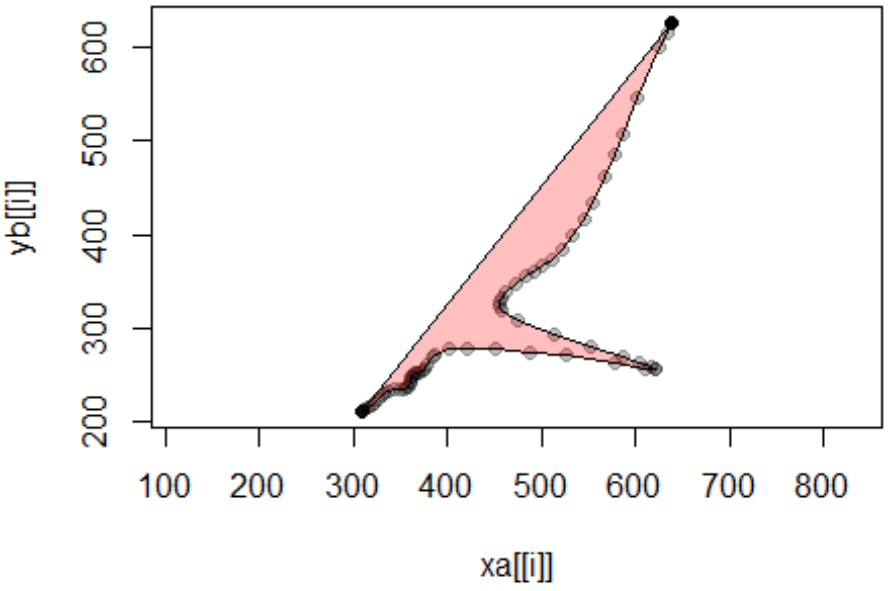

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

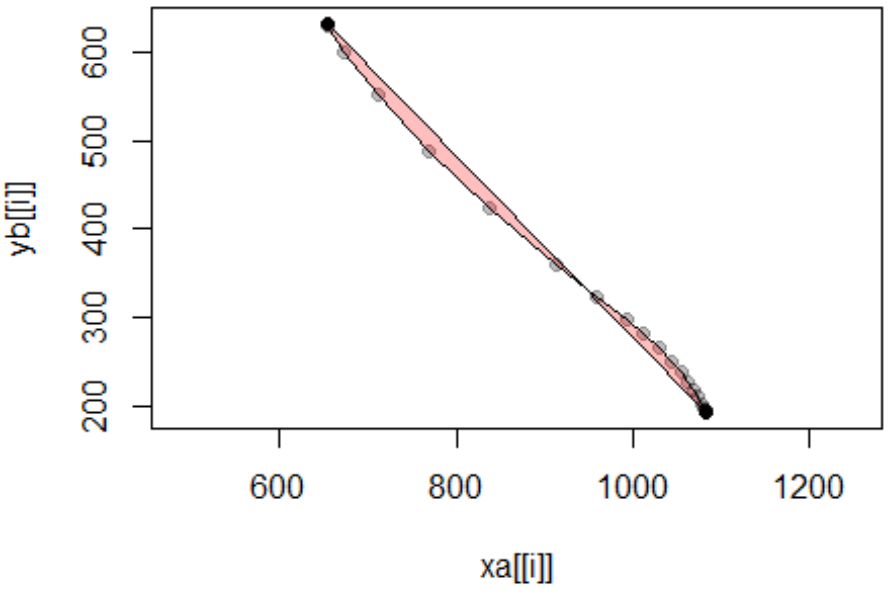

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

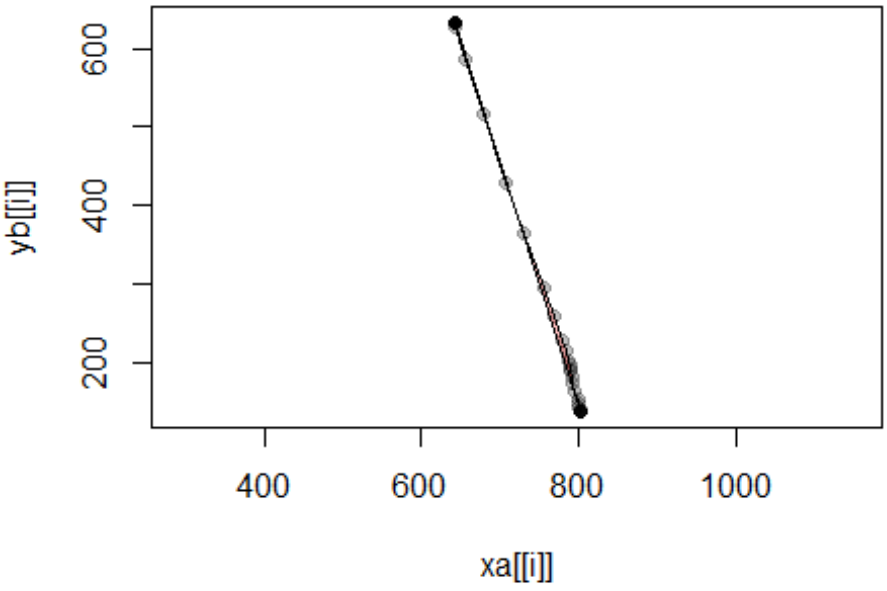

yb[[i]]
300
400
500
600
100
200
300
400
500
600
700
800
xa[[i]]

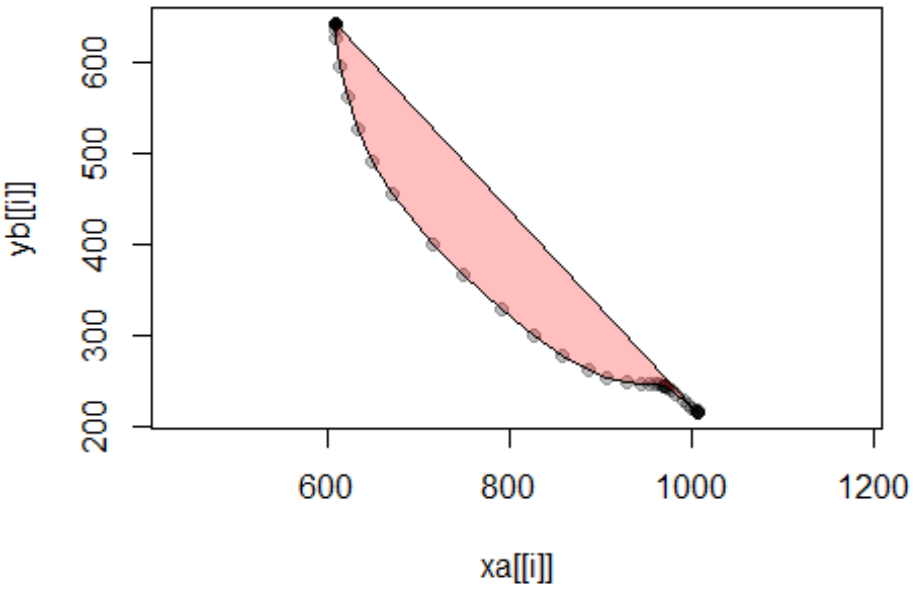

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

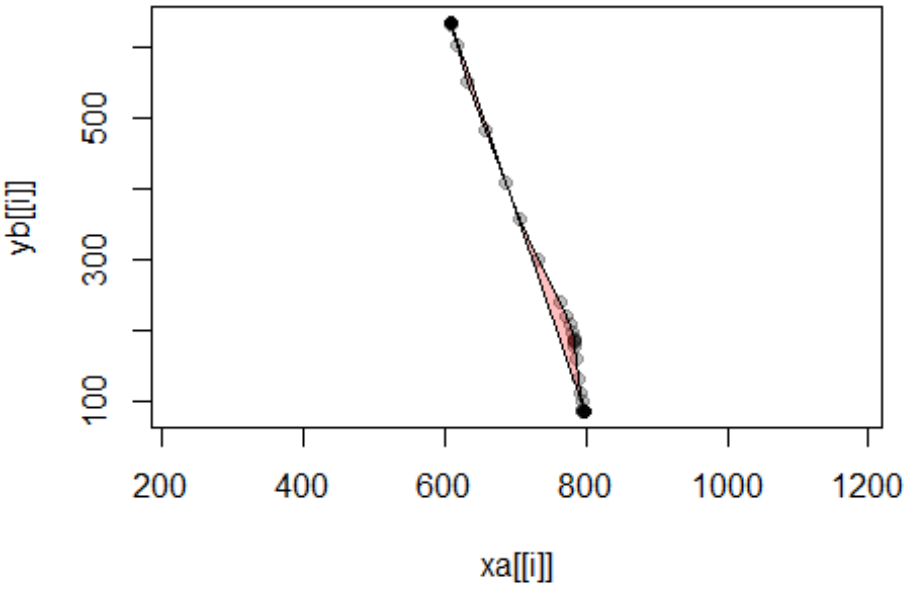

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

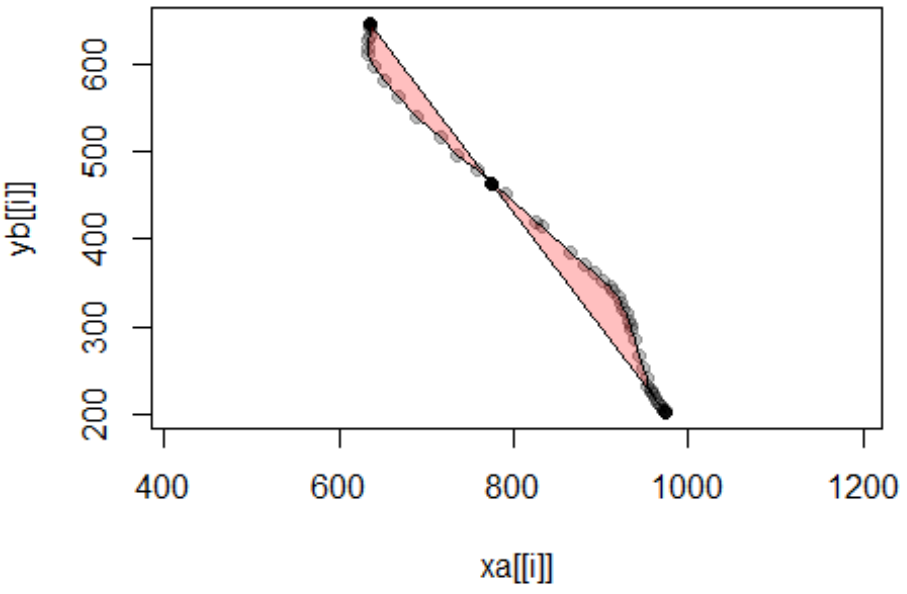

yb[[i]]
200
300
400
500
600
100
200
300
400
500
600
700
800
xa[[i]]

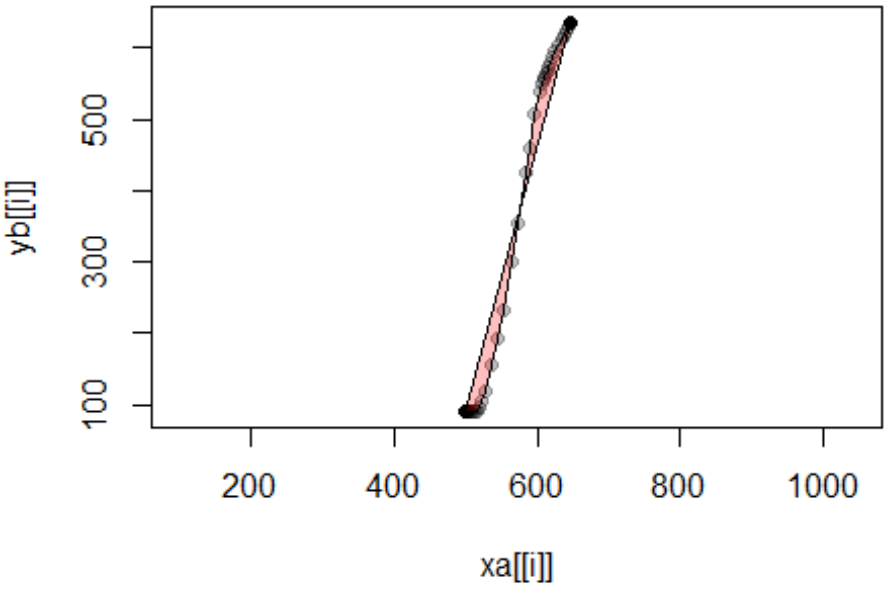

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

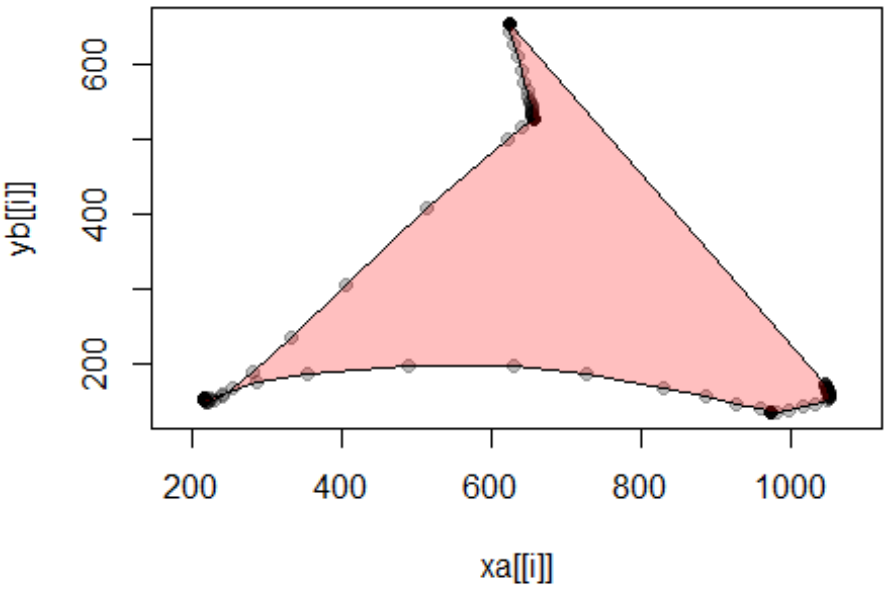

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

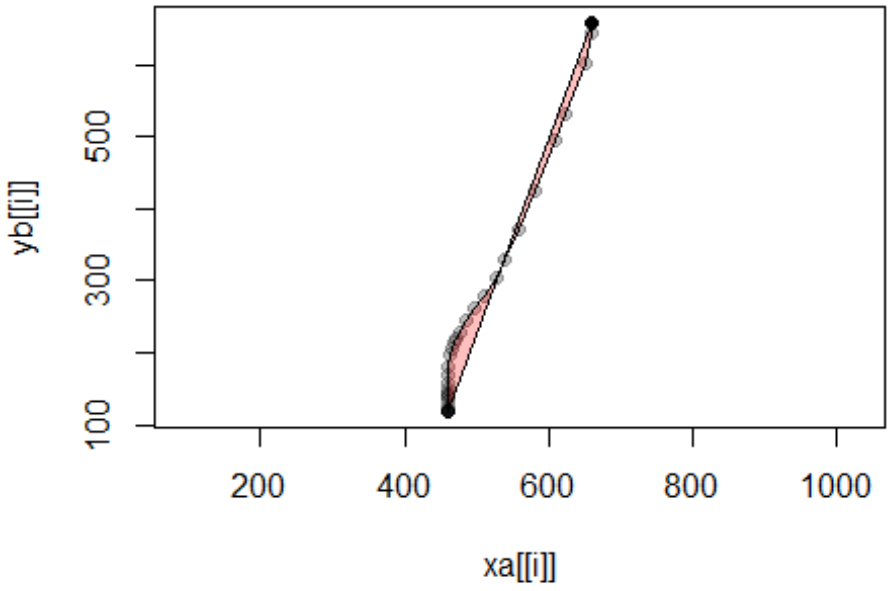

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

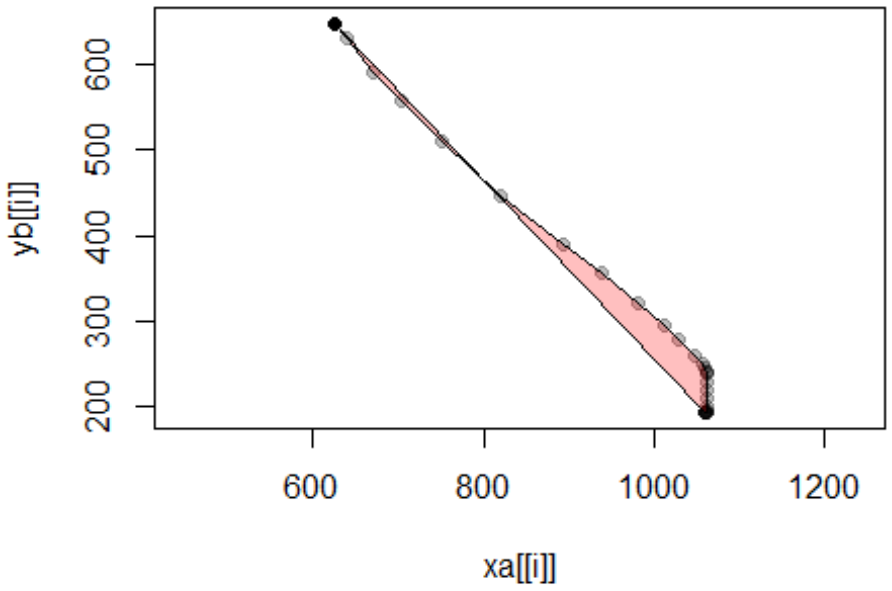

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

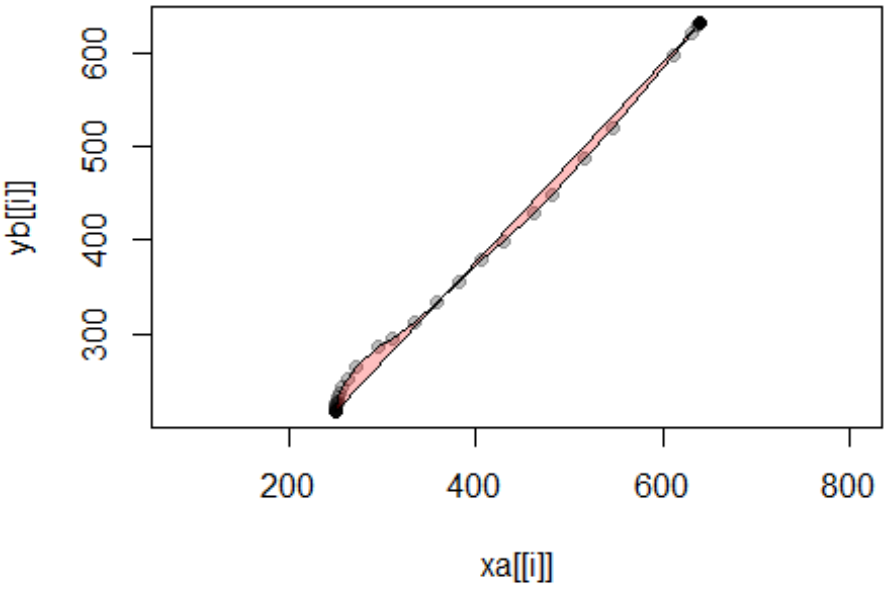

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

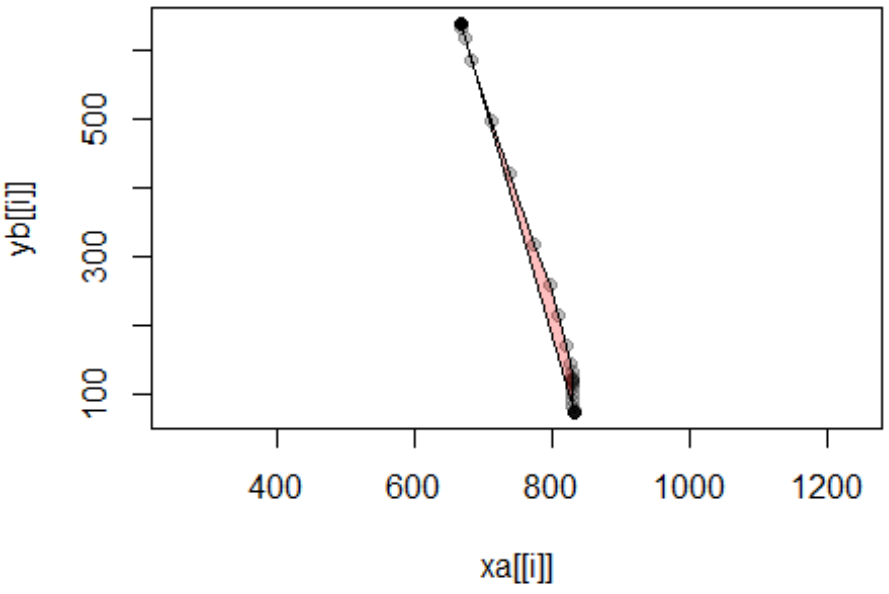

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

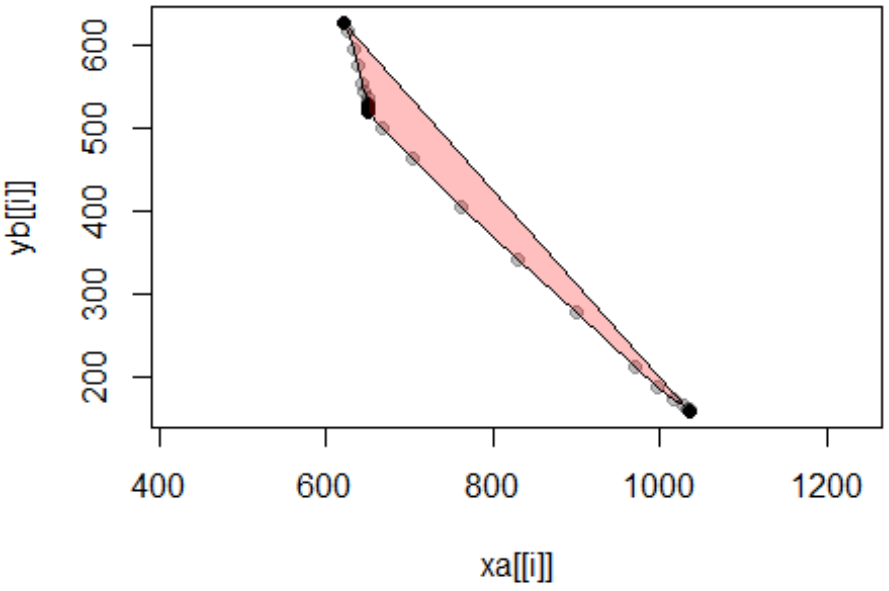

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

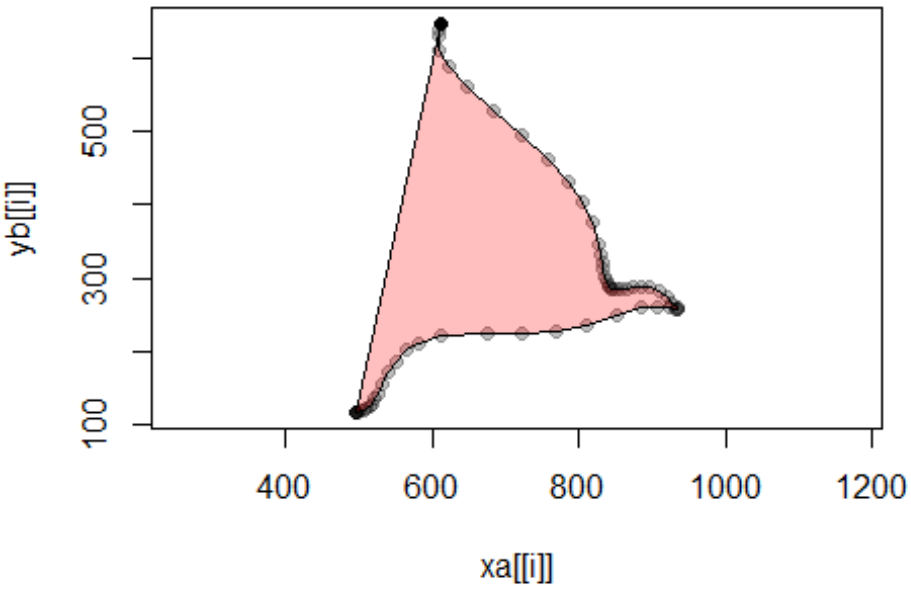

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

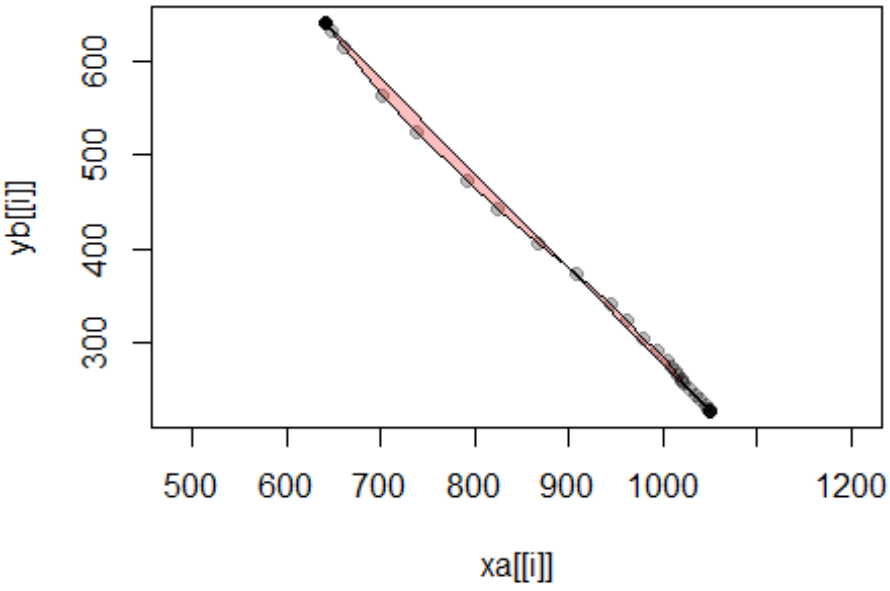

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

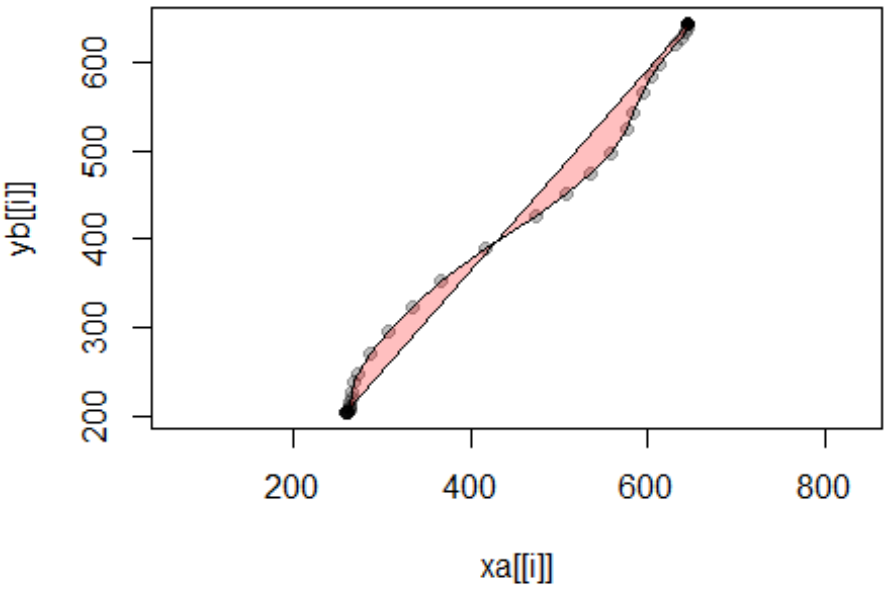

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

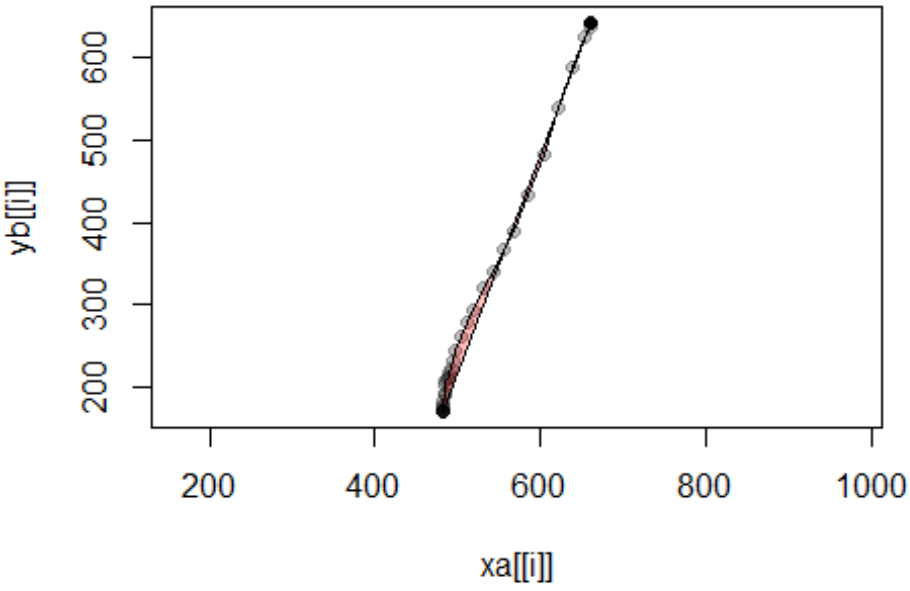

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

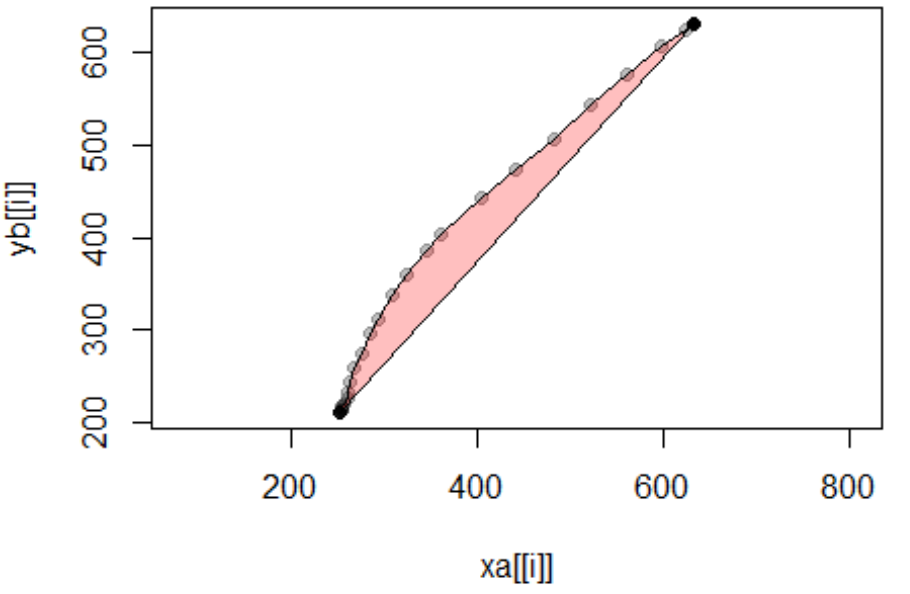

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

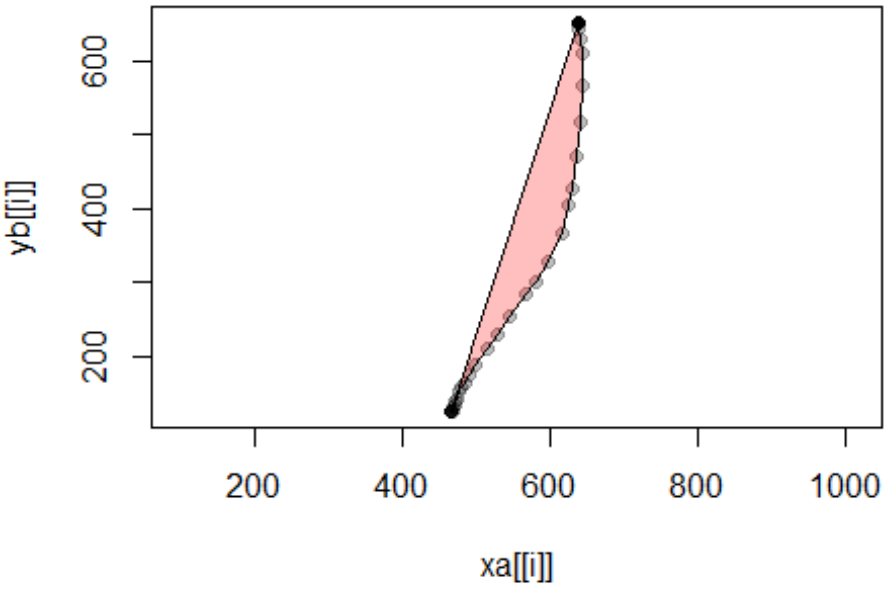

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

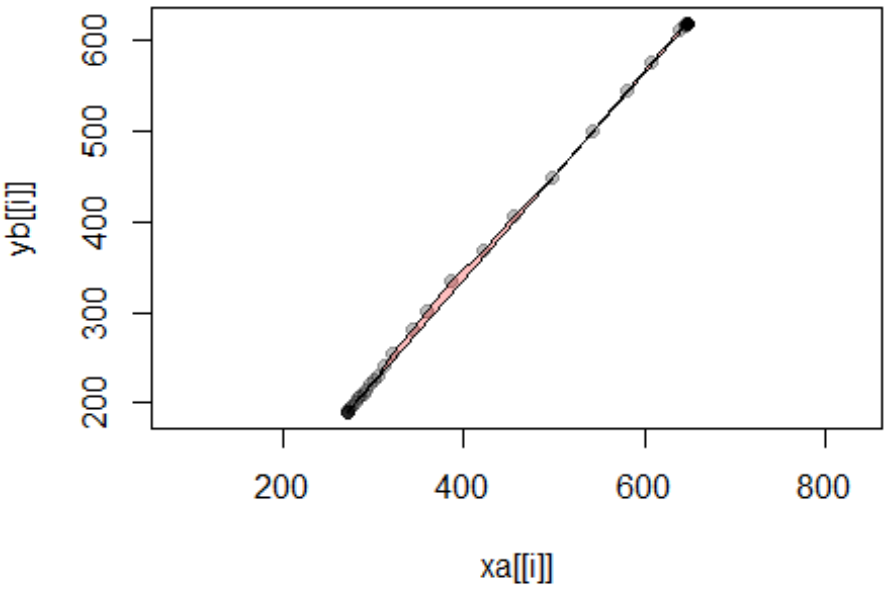

yb[[i]]
300
400
500
600
500
600
700
800
900
1000
1200
xa[[i]]

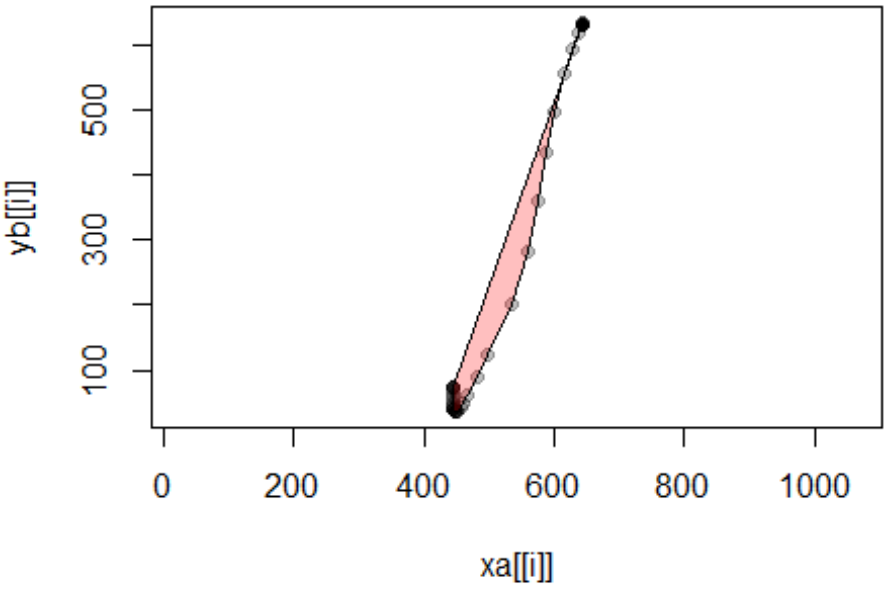

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

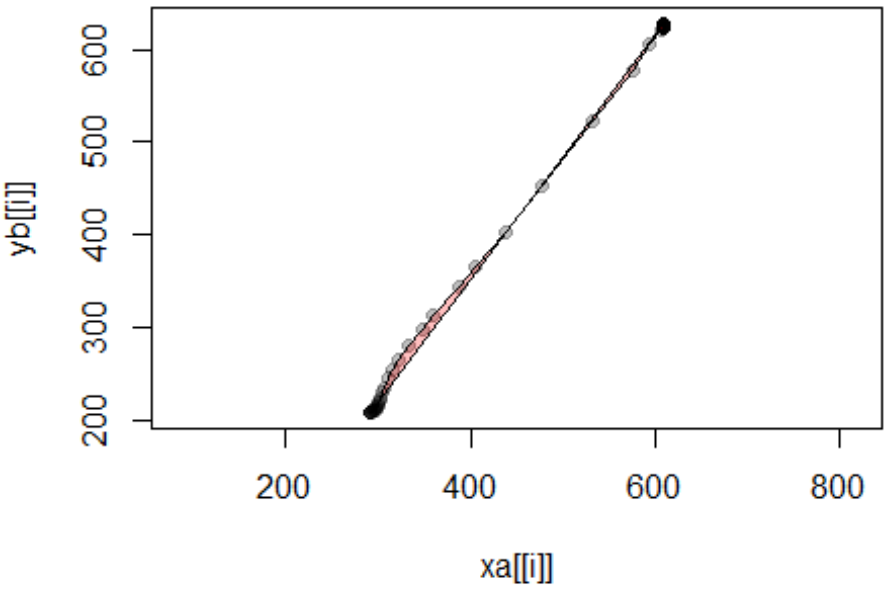

yb[[i]]
200
300
400
500
600
200
400
600
800
1000
xa[[i]]

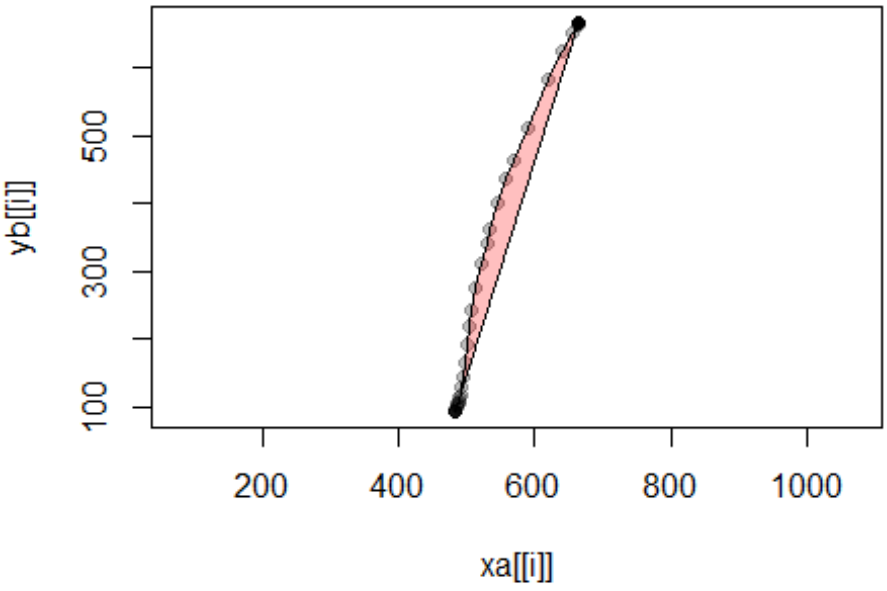

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

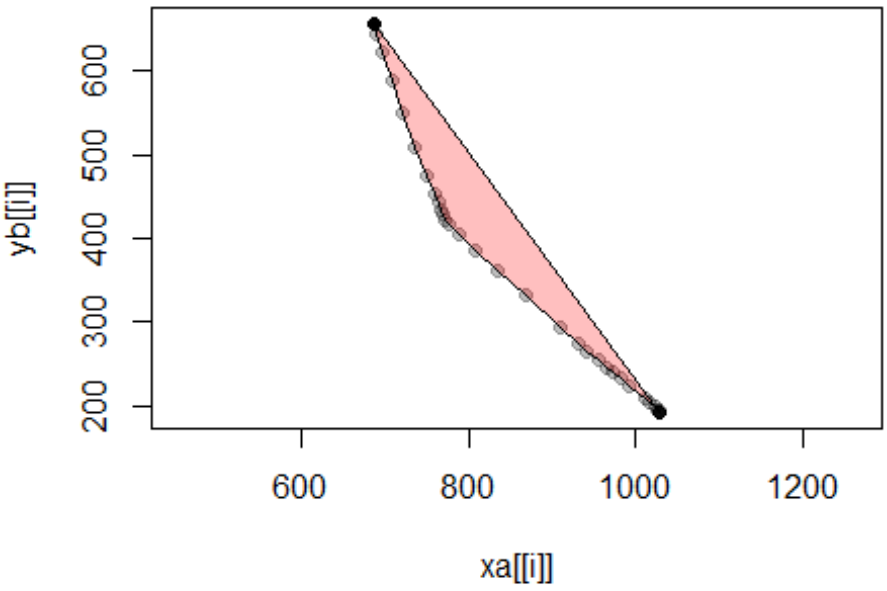

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

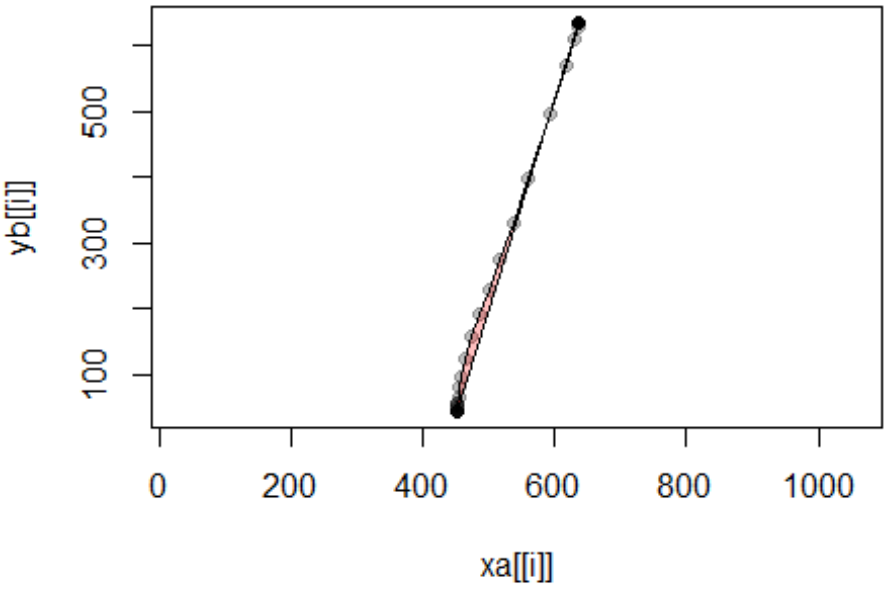

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

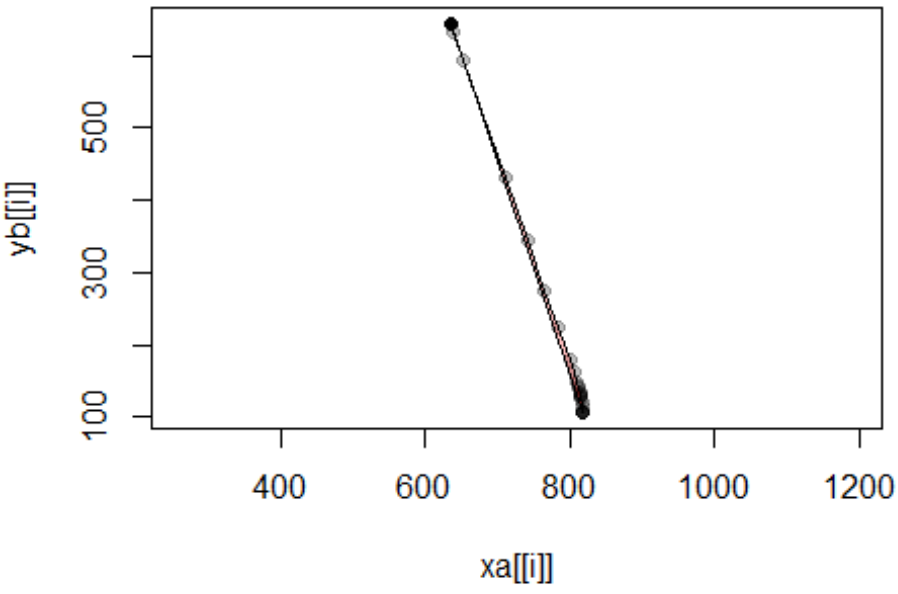

yb[[i]]
100
300
500
0
200
400
600
800
1000
xa[[i]]

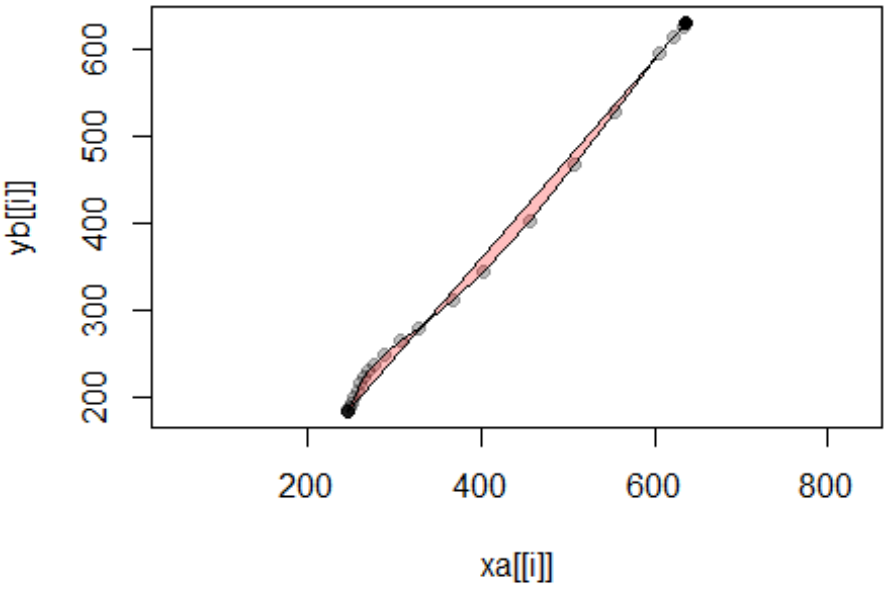

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

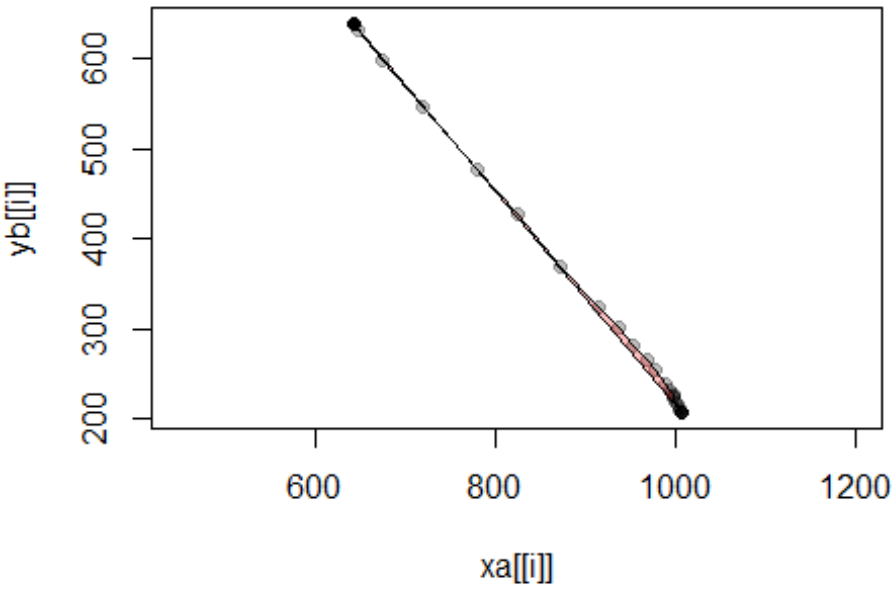

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

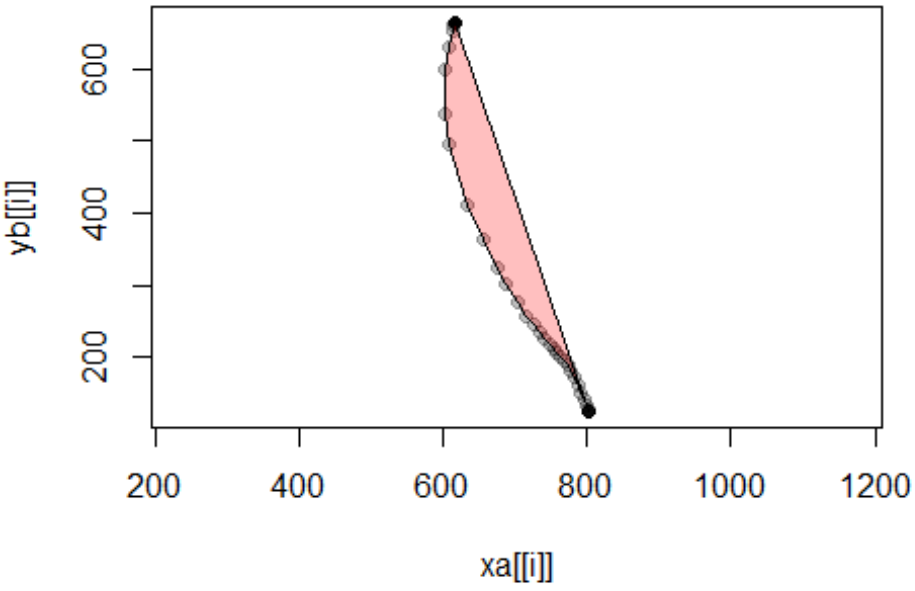

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

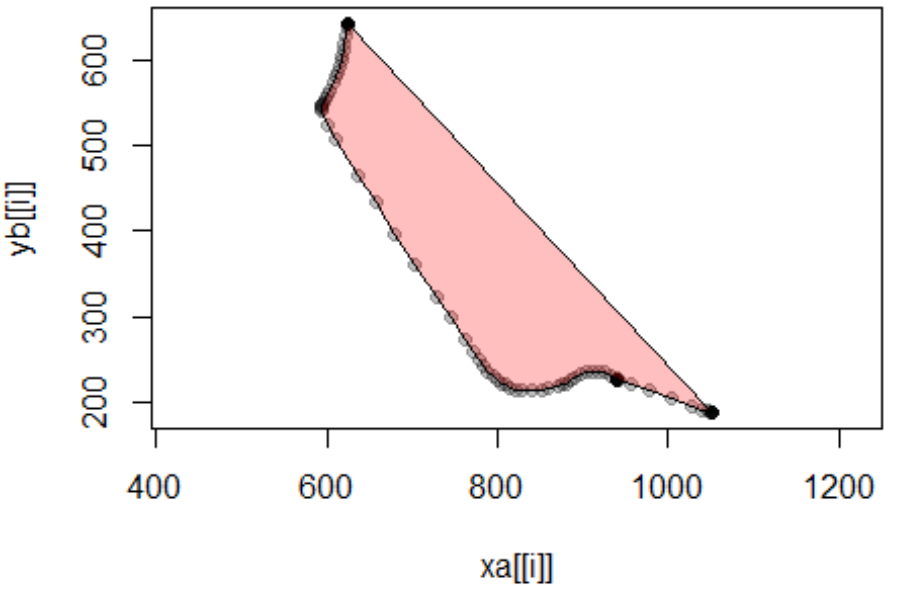

yb[[i]]
100
300
500
0
200
400
600
800
1000
xa[[i]]

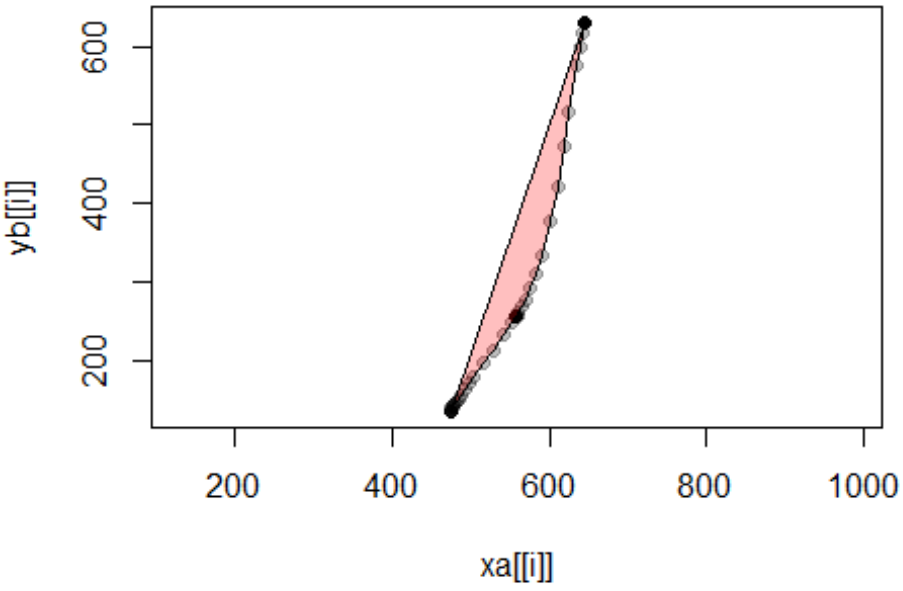

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

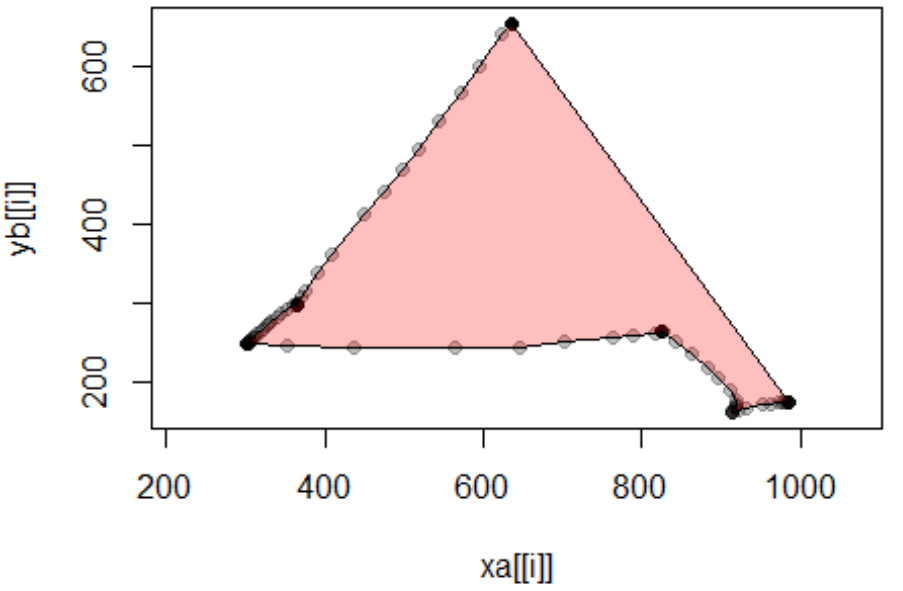

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

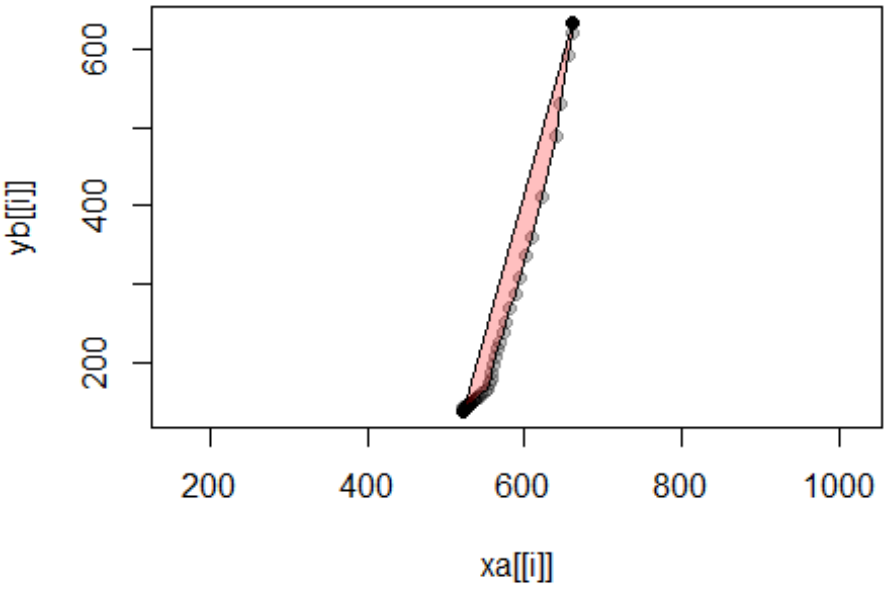

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

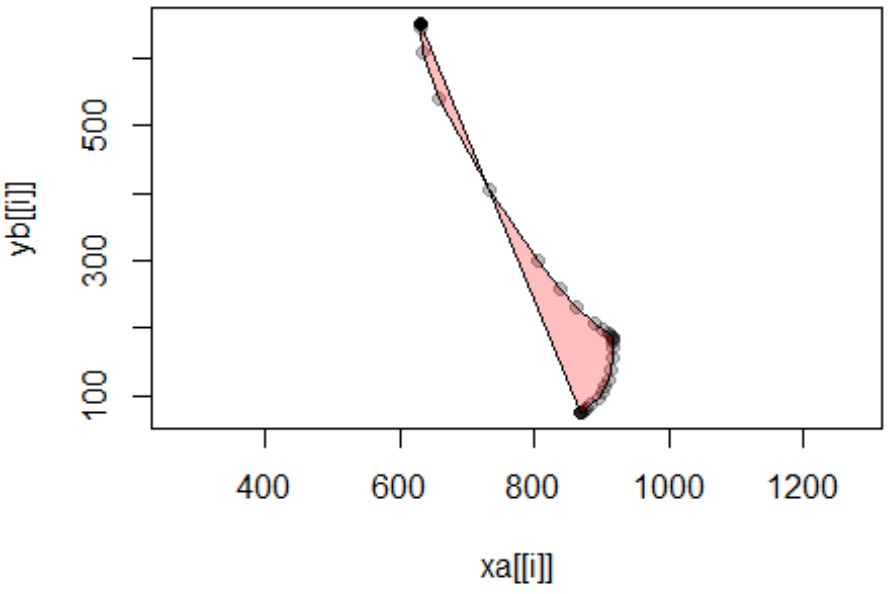

yb[[i]]
200
400
600
200
400
600
800
1000
1200
xa[[i]]

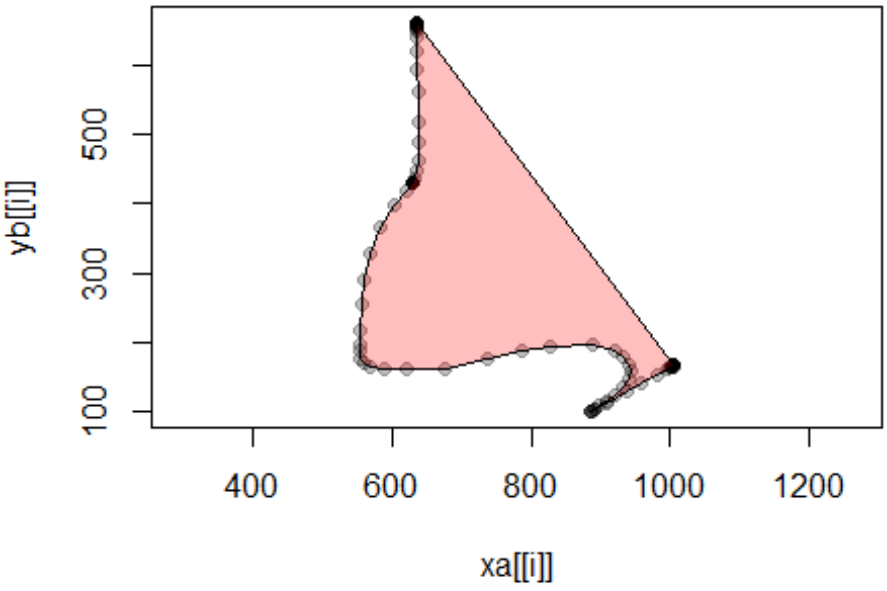

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

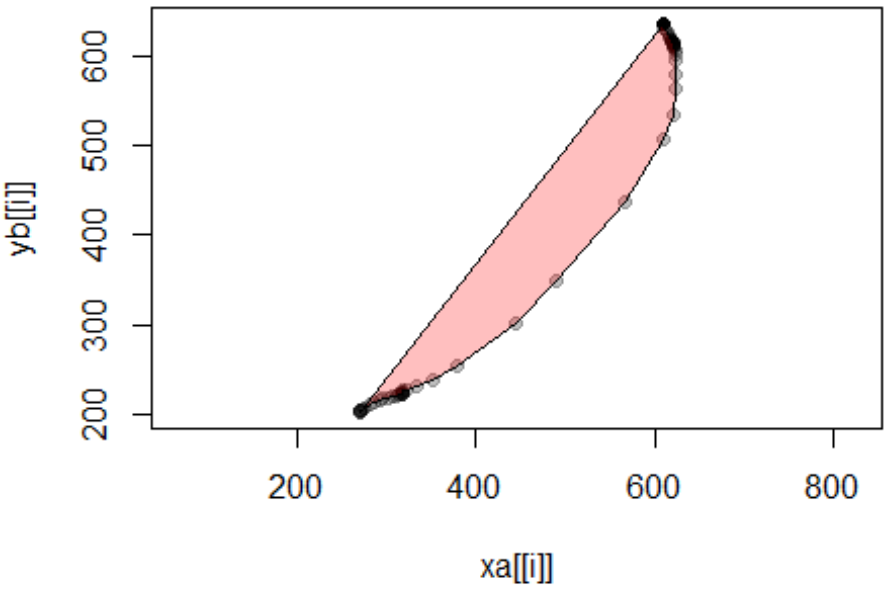

yb[[i]]
600
400
200
200
400
600
800
1000
xa[[i]]

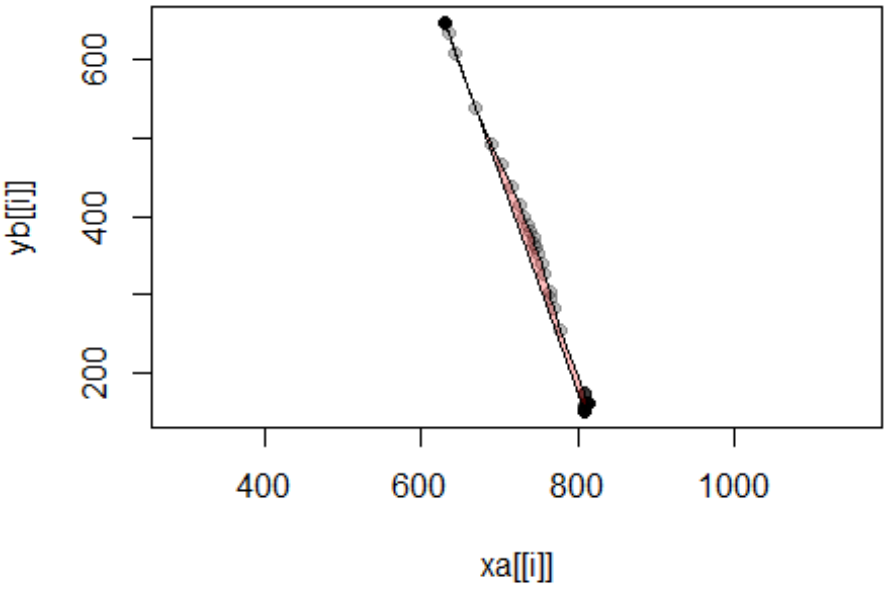

yb[[i]]
600
400
200
200
400
600
800
1000
xa[[i]]

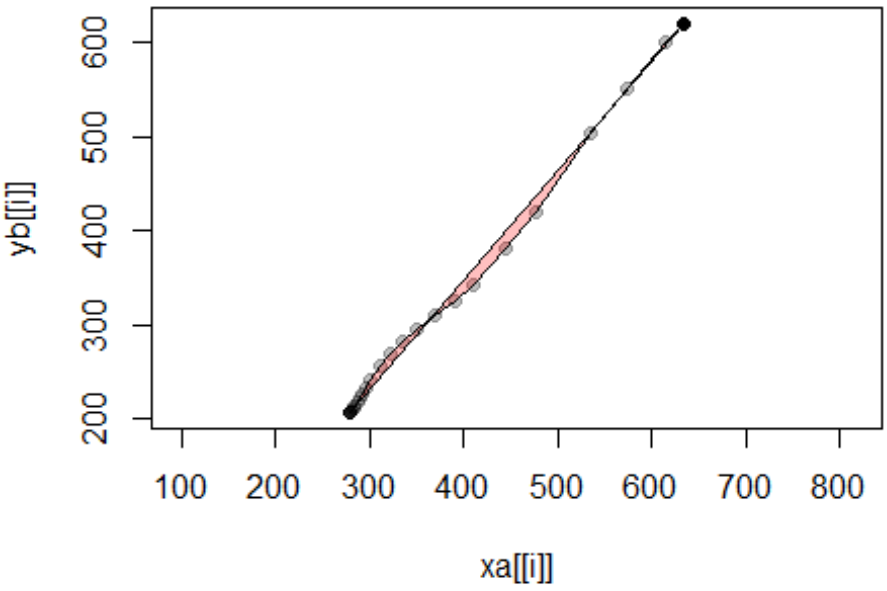

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

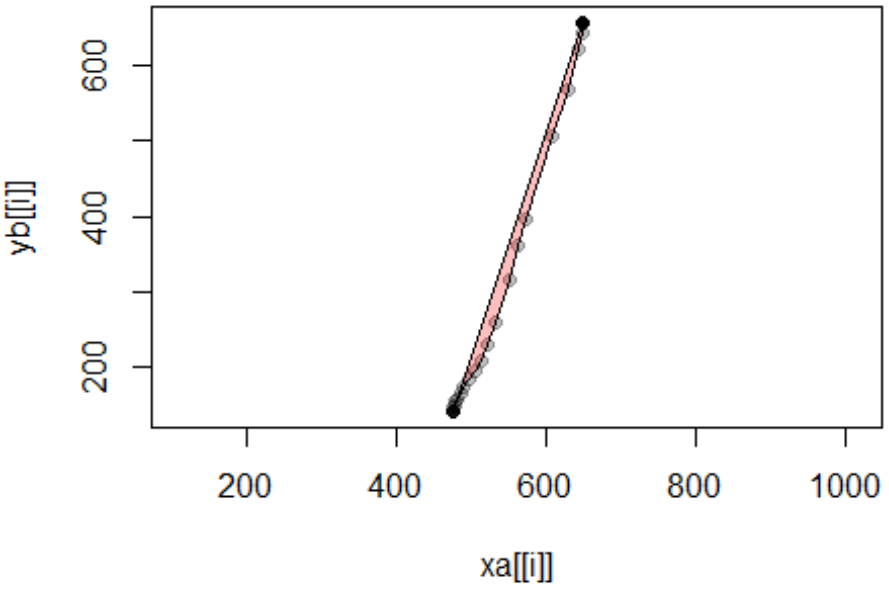

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

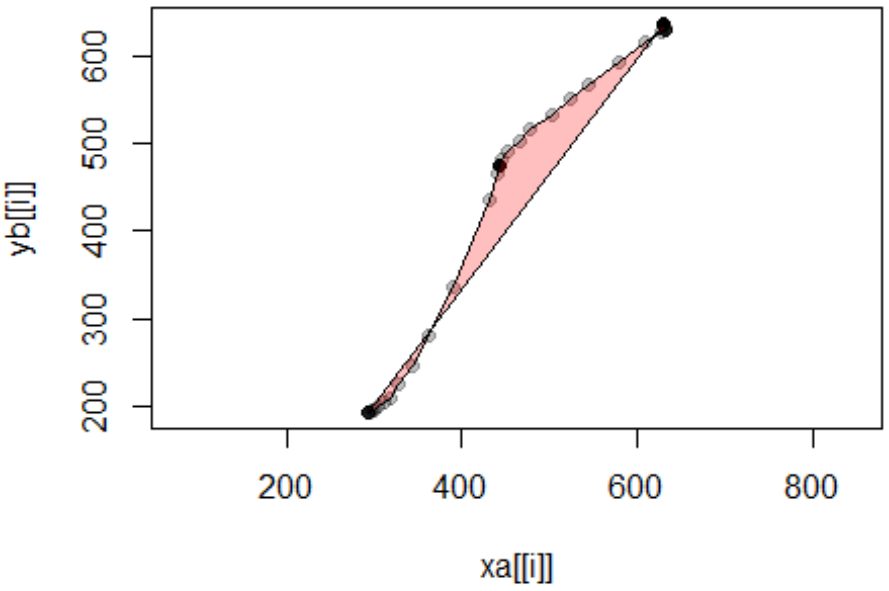

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

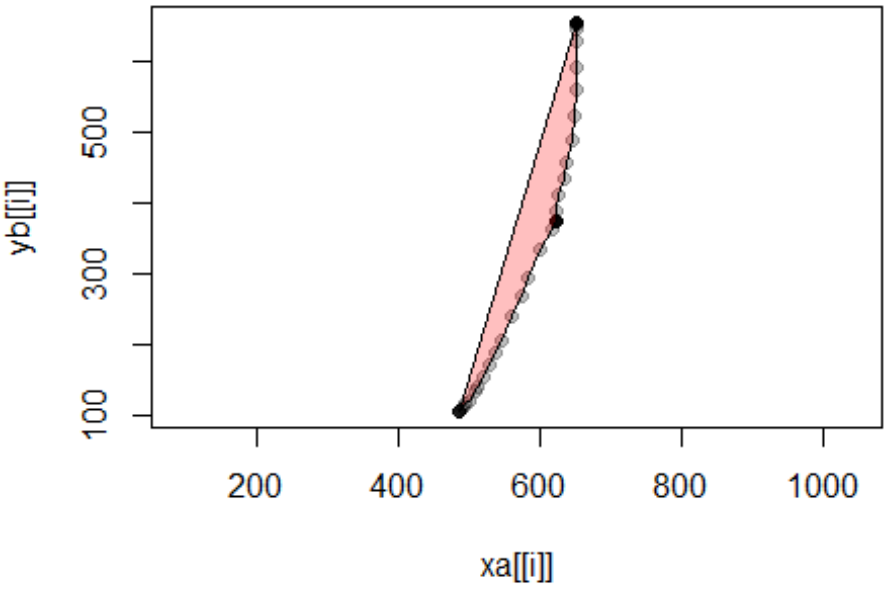

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

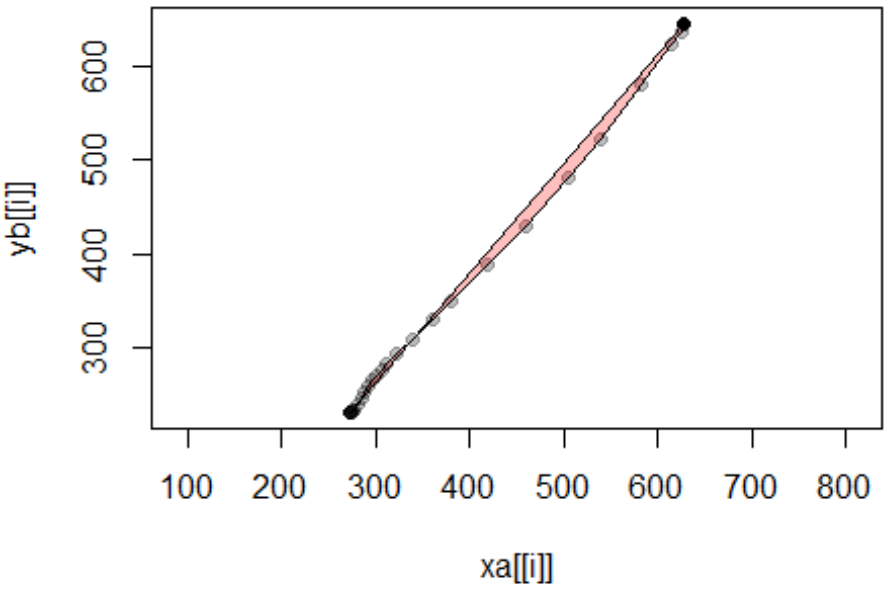

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

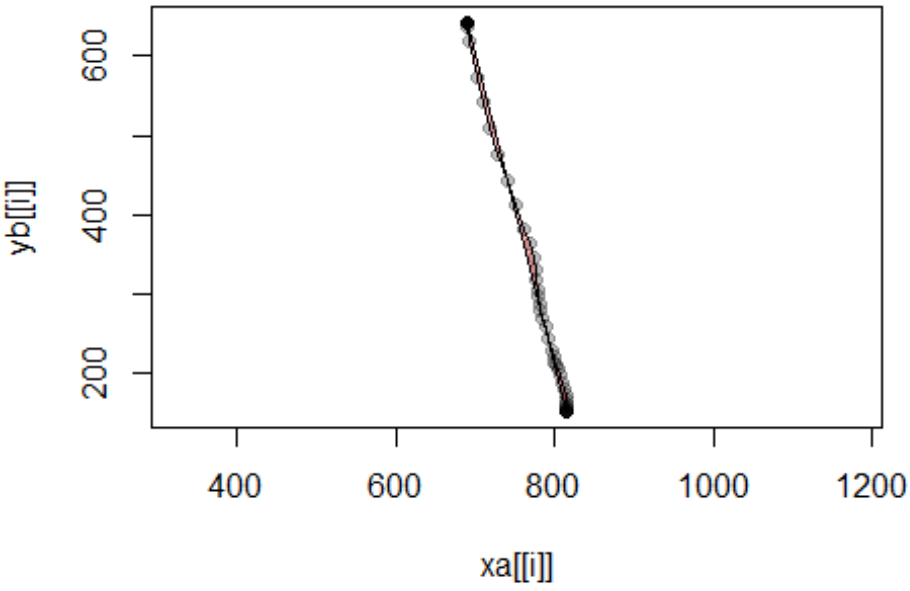

yb[[i]]
200
300
400
500
600
100
200
300
400
500
600
700
800
xa[[i]]

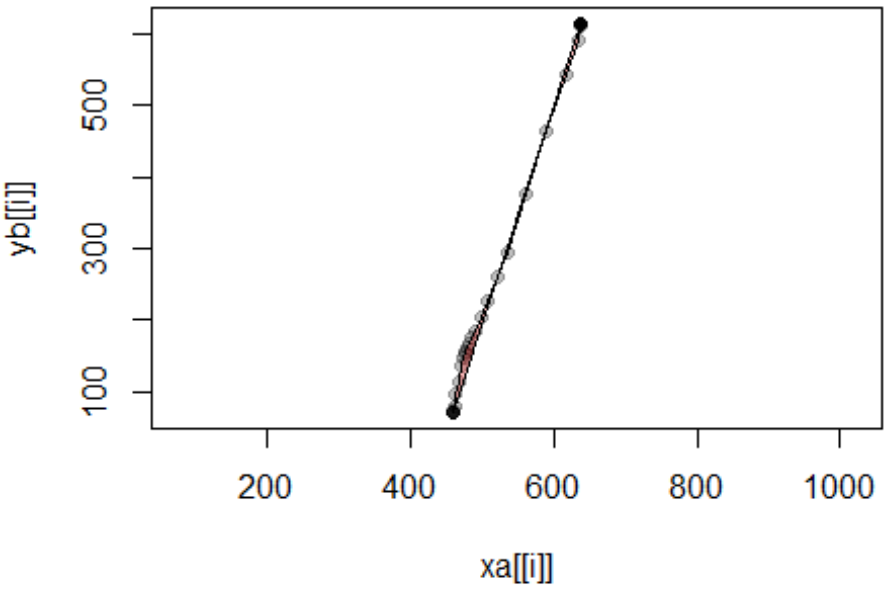

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

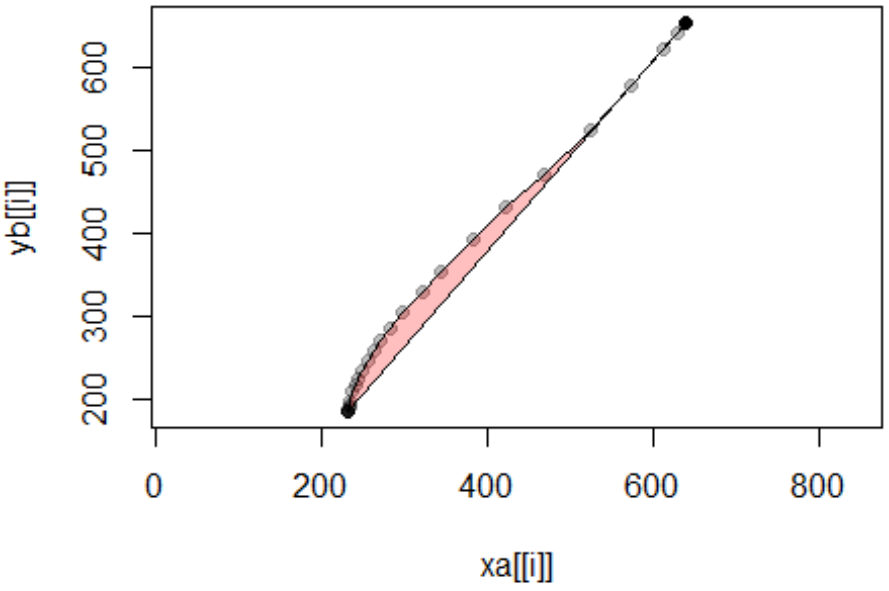

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

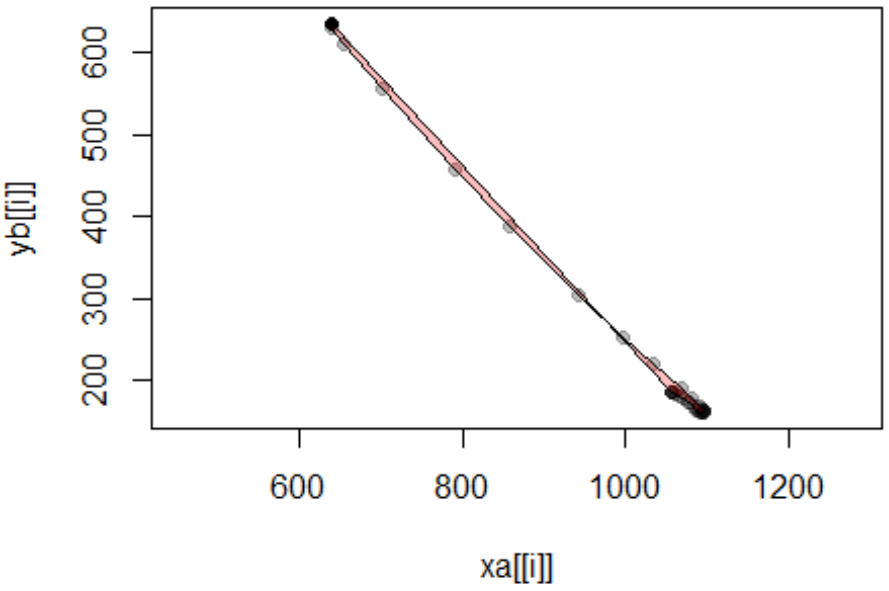

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

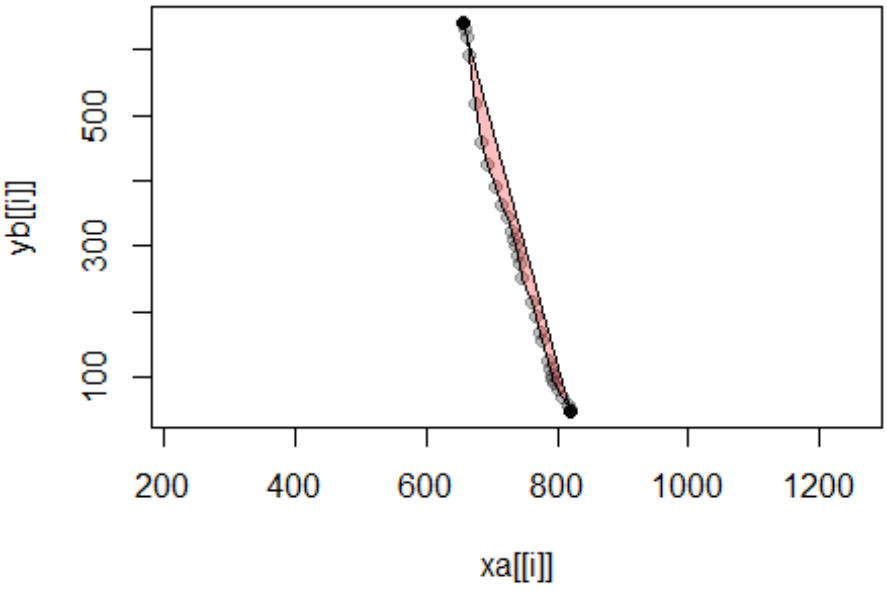

yb[[i]]
300
400
500
600
100
200
300
400
500
600
700
800
xa[[i]]

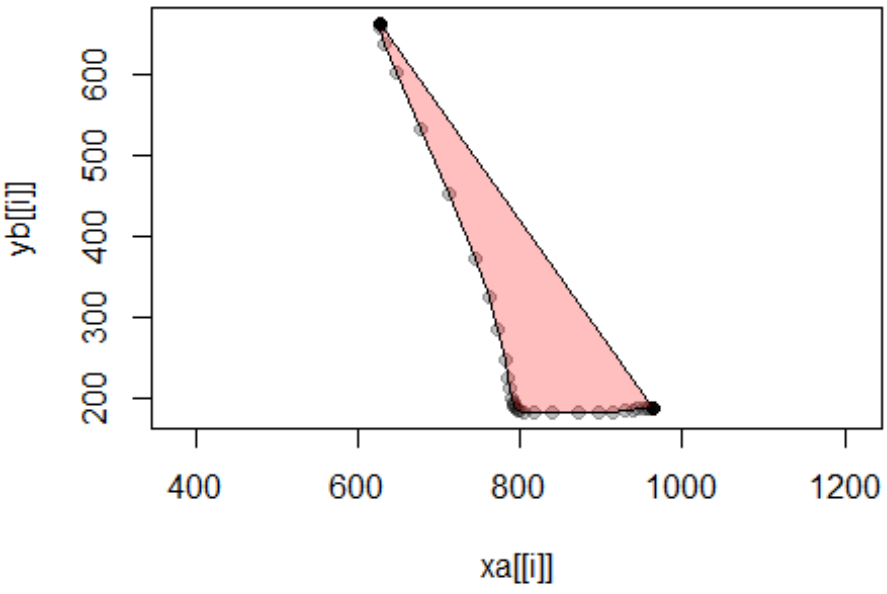

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

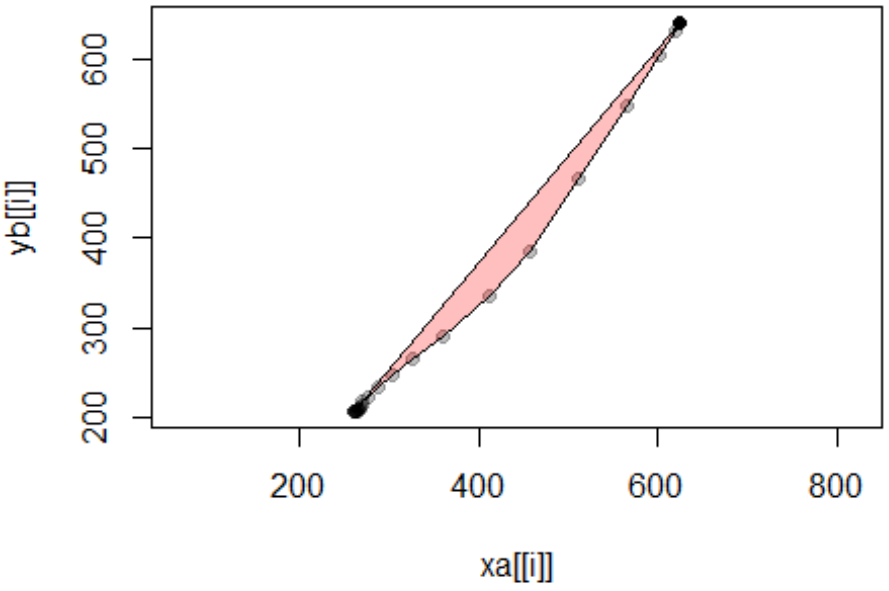

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

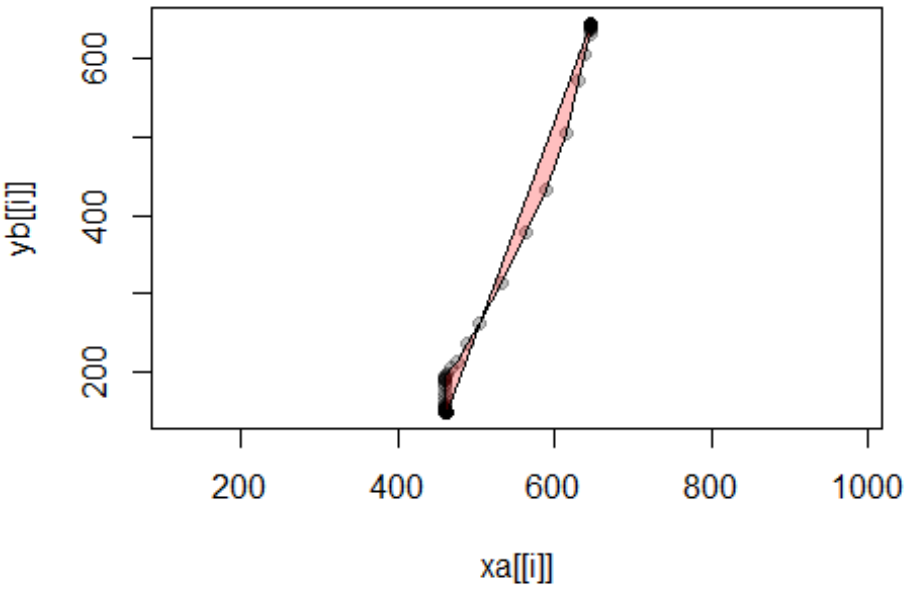

yb[[i]]
200
300
400
500
600
0
200
400
600
800
xa[[i]]

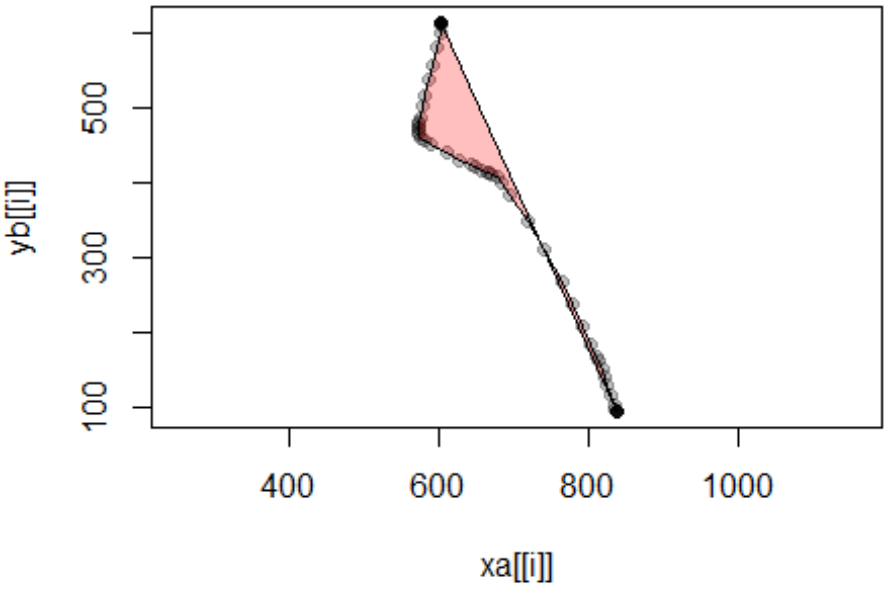

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

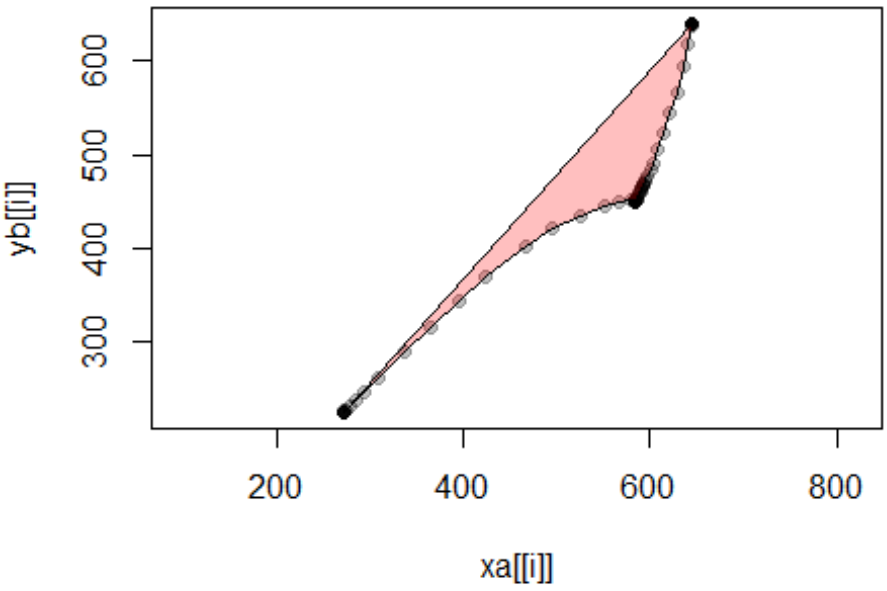

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

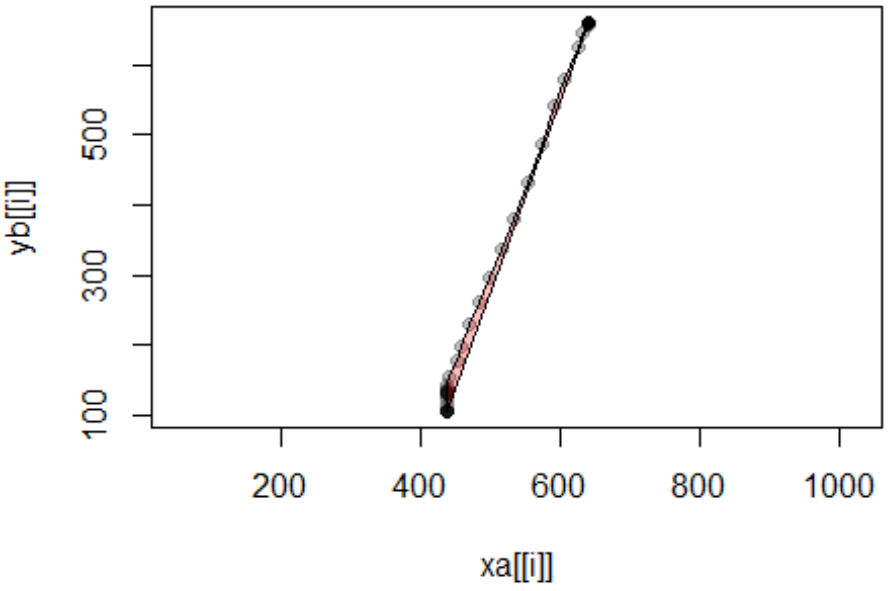

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

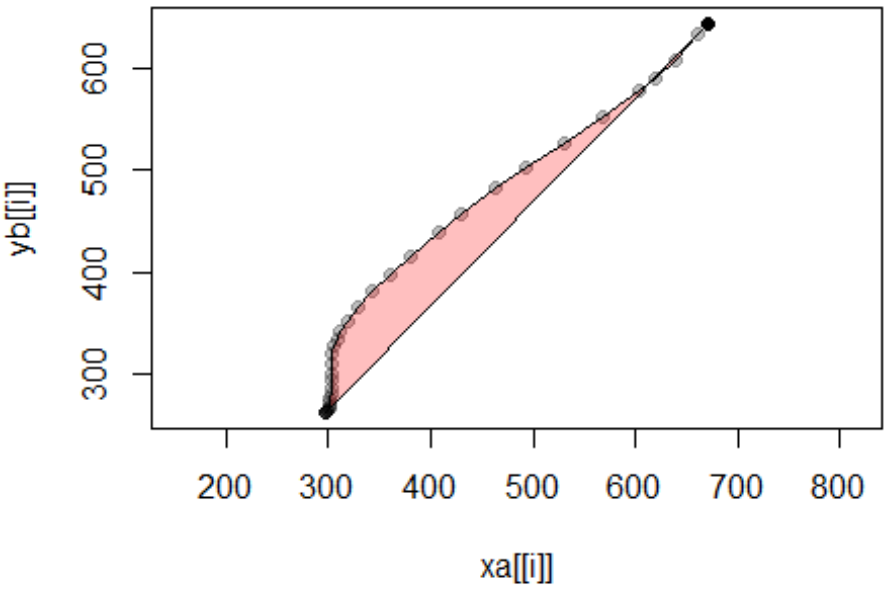

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

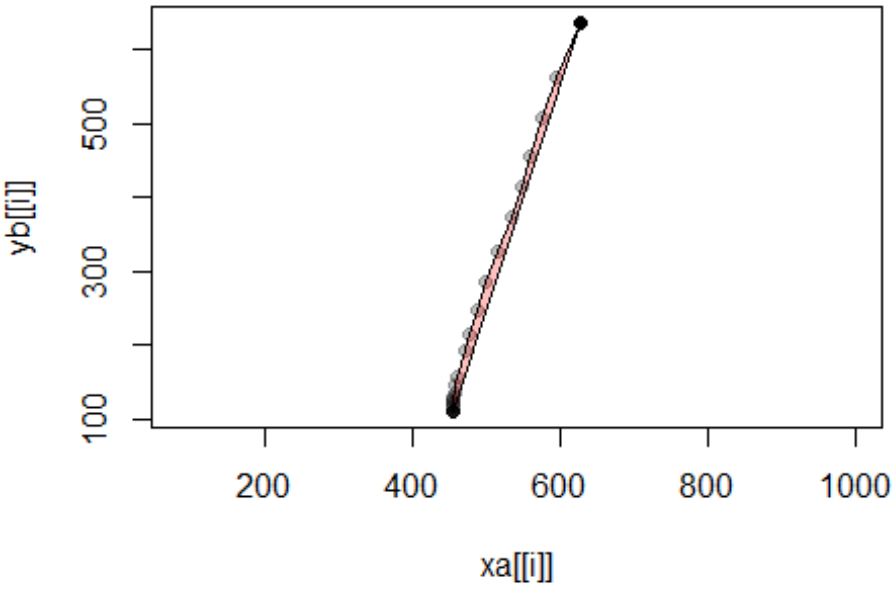

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

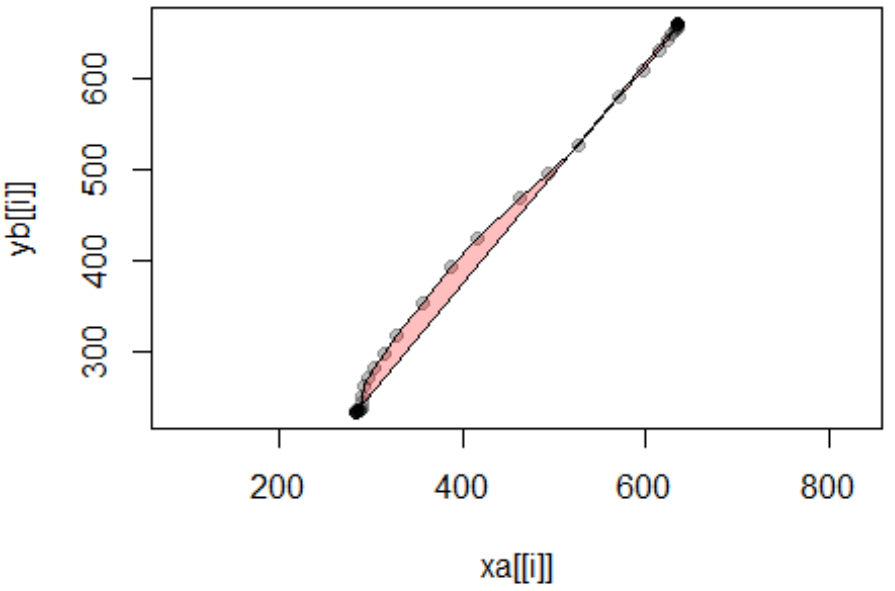

yb[[i]]
100
300
500
400
600
800
1000
xa[[i]]

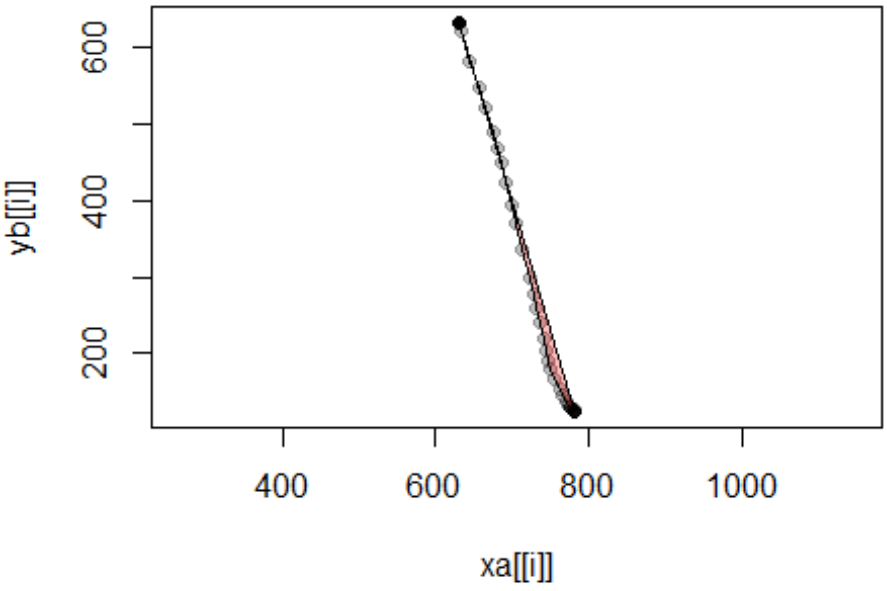

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

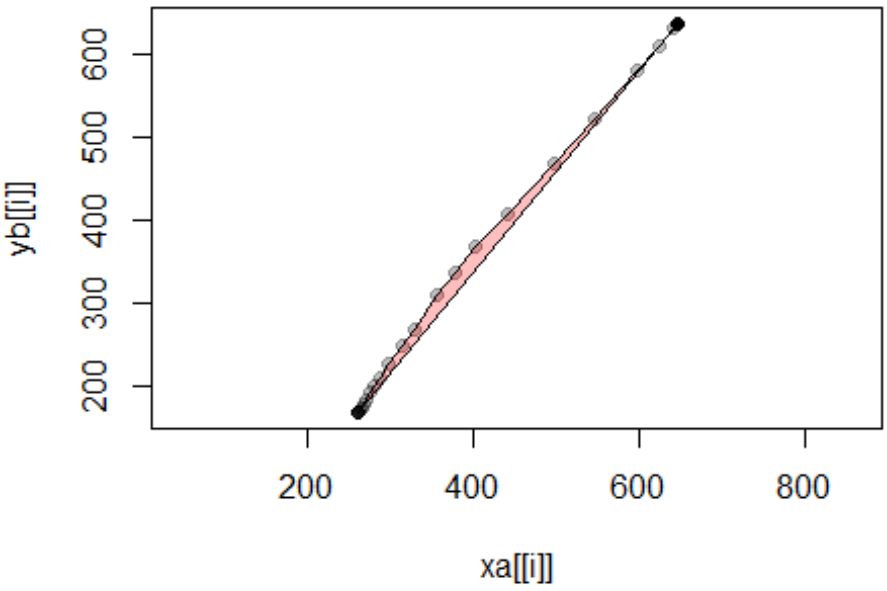

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

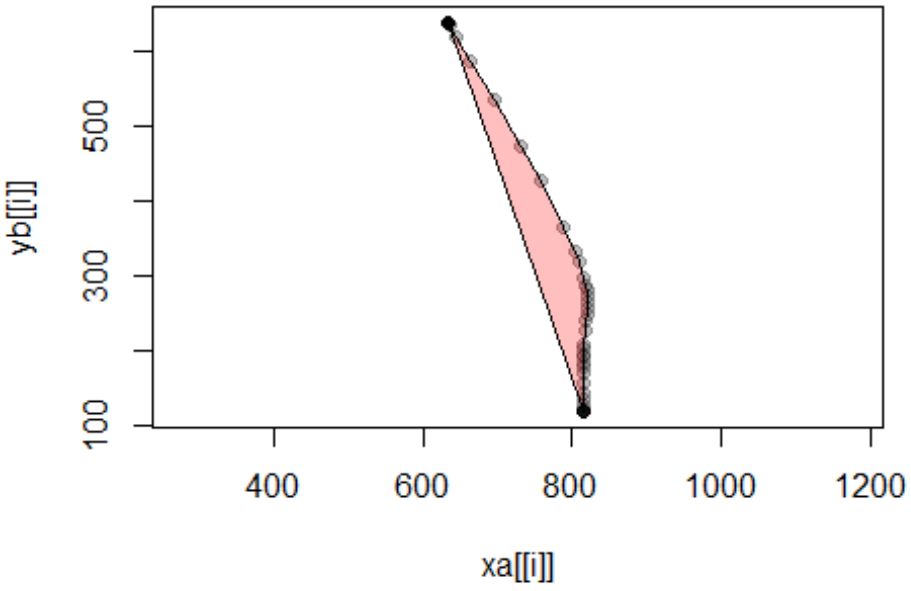

yb[[i]]
300
400
500
600
200
300
400
500
600
700
800
xa[[i]]

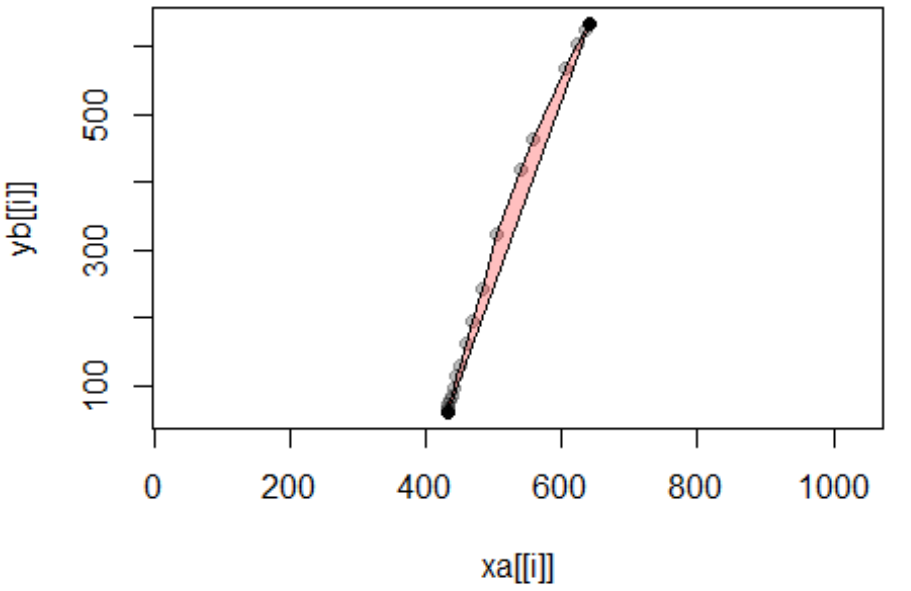

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

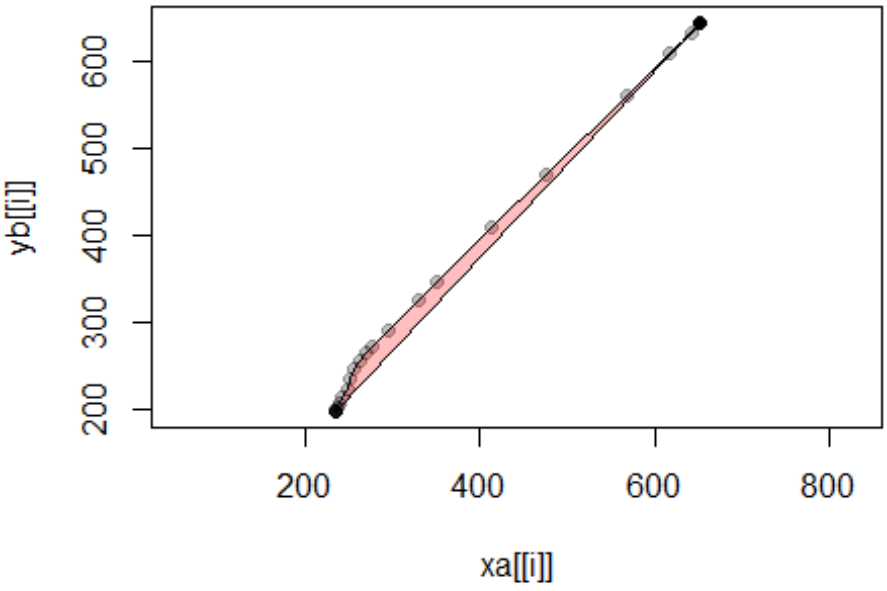

yb[[i]]
300
400
500
600
200
400
600
800
xa[[i]]

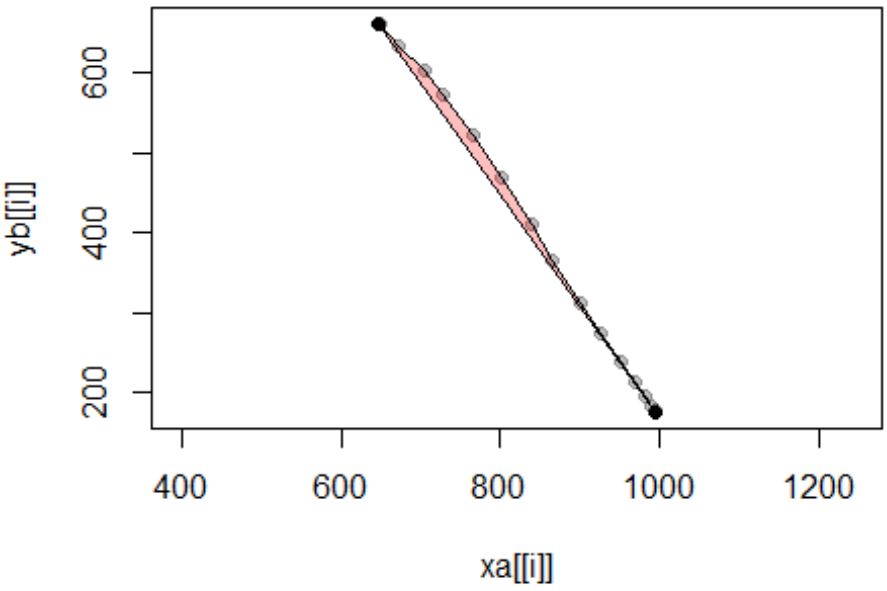

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

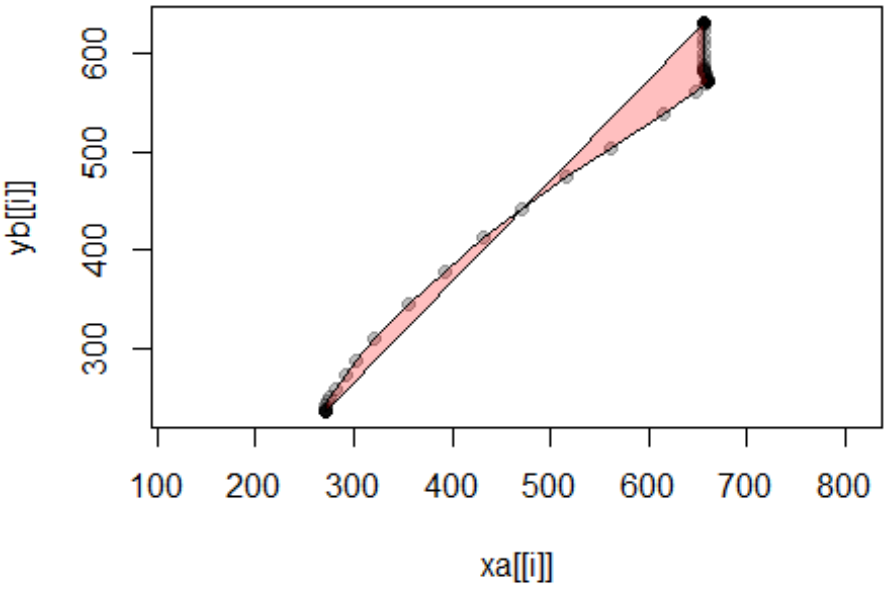

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

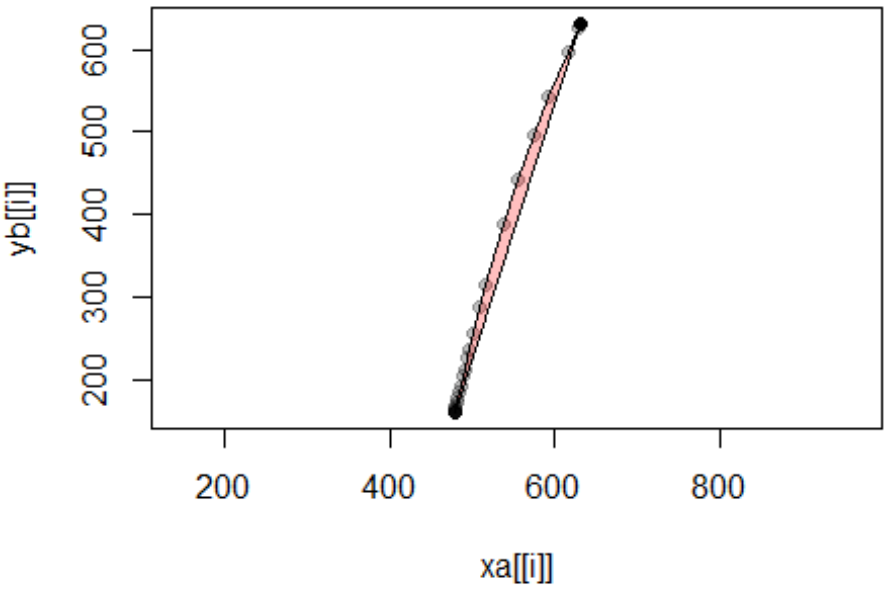

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

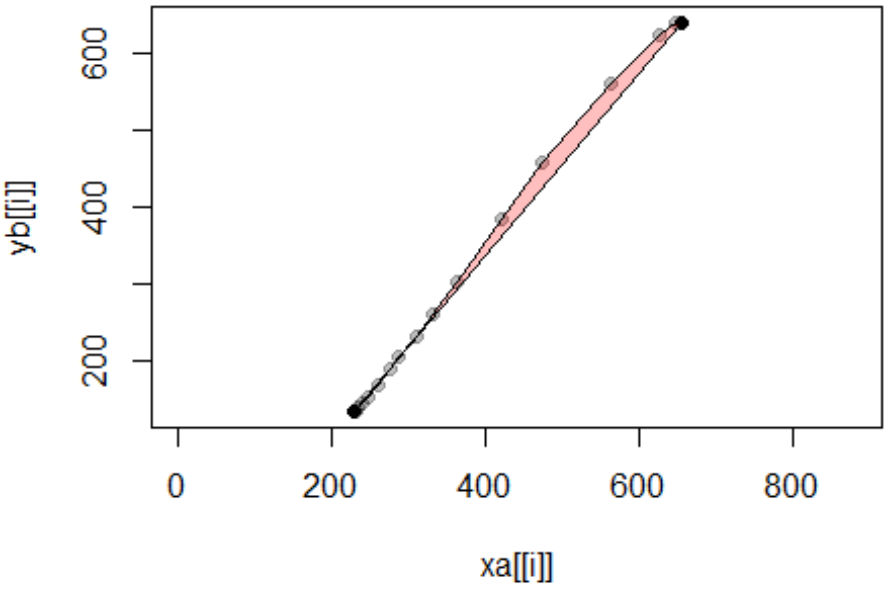

yb[[i]]
100
300
500
0
200
400
600
800
1000
xa[[i]]

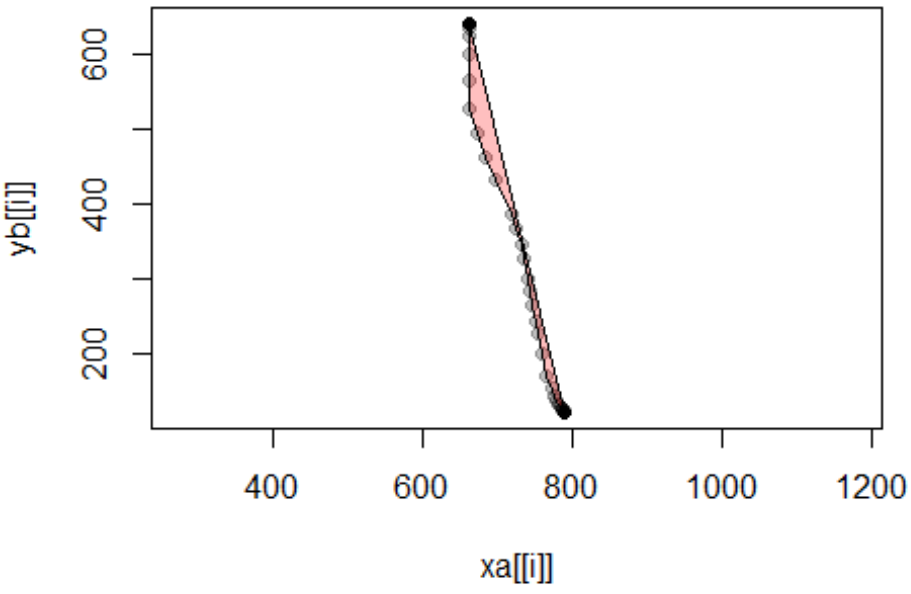

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

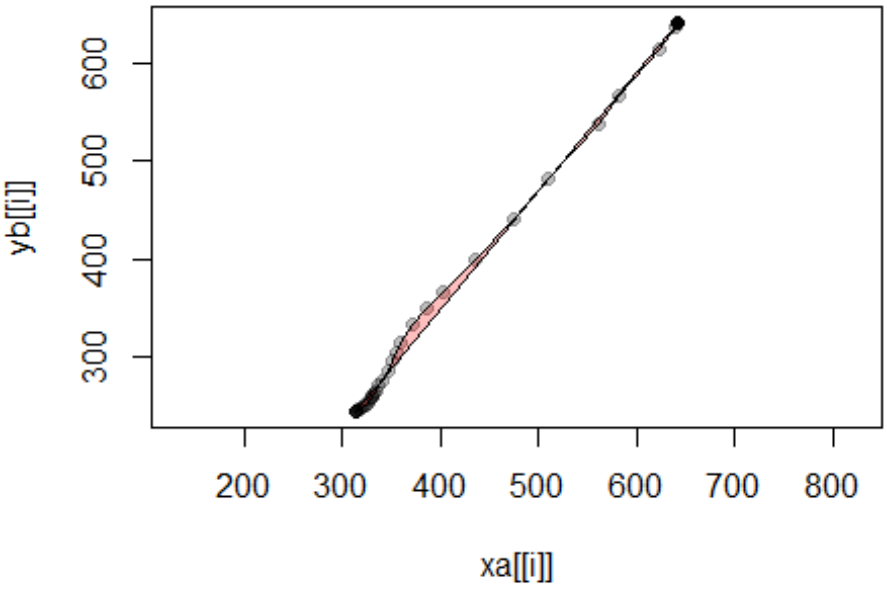

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

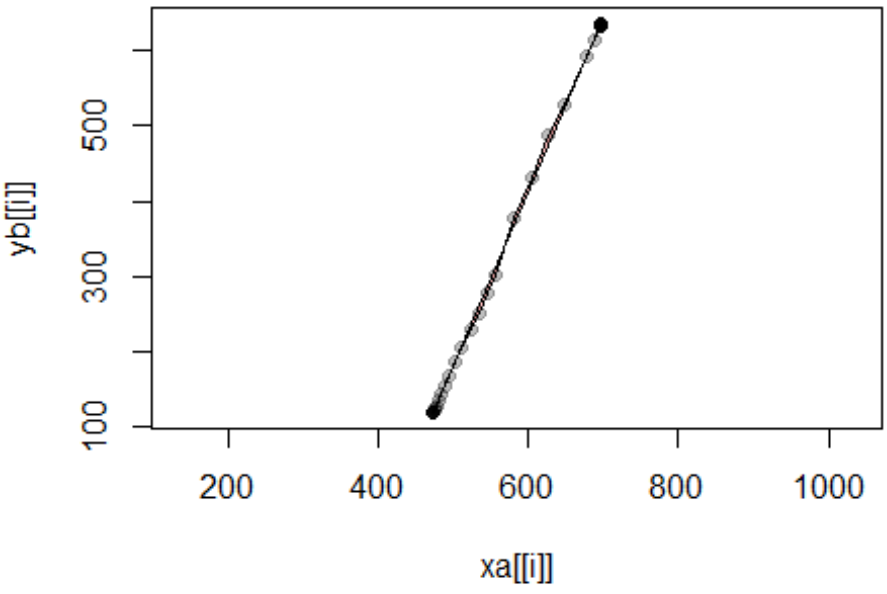

yb[[i]]
300
400
500
600
100
200
300
400
500
600
700
800
xa[[i]]

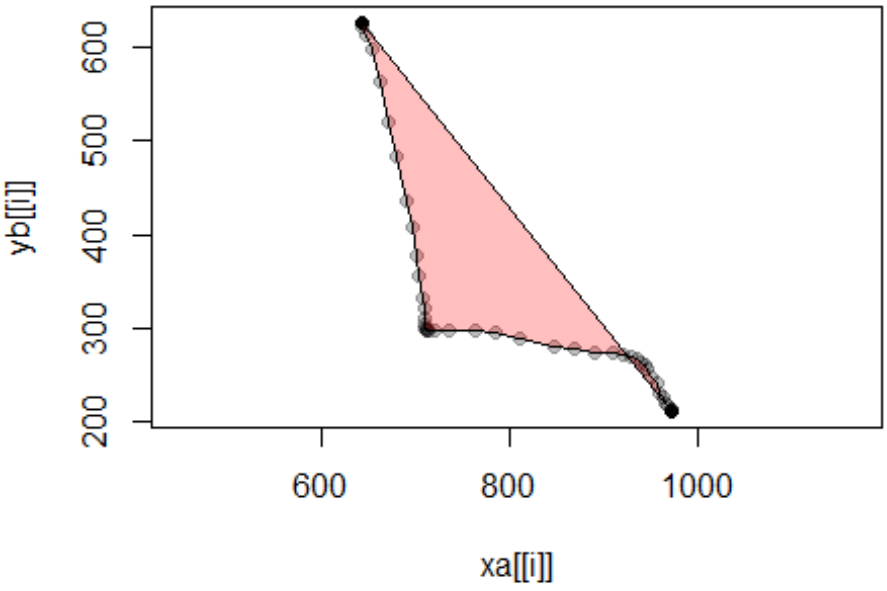

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

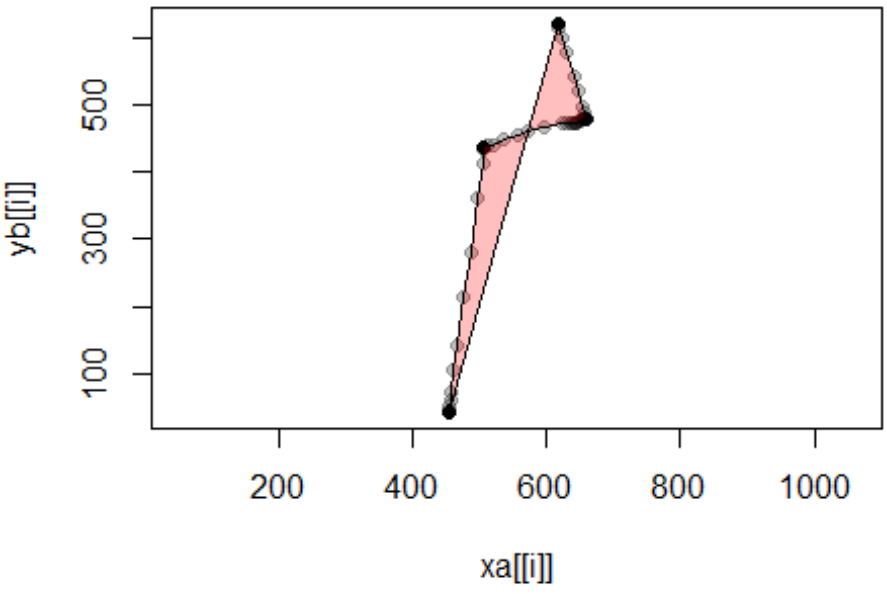

yb[[i]]
200
400
600
0
200
400
600
800
xa[[i]]

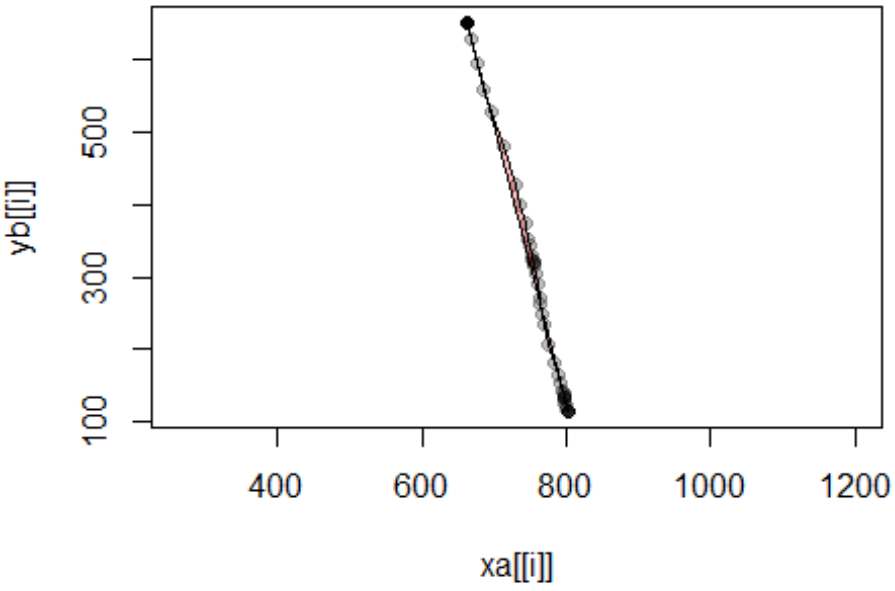

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

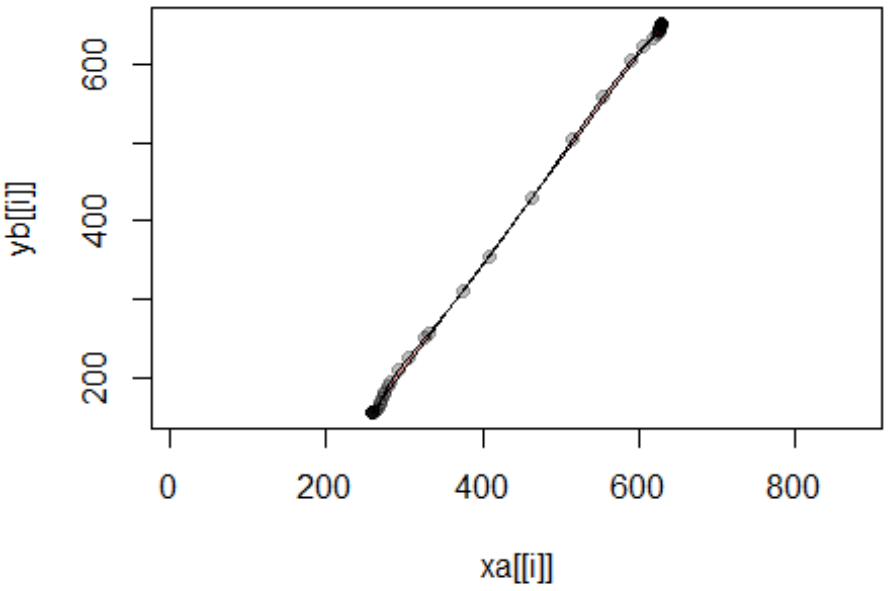

yb[[i]]
300
400
500
600
200
300
400
500
600
700
800
xa[[i]]

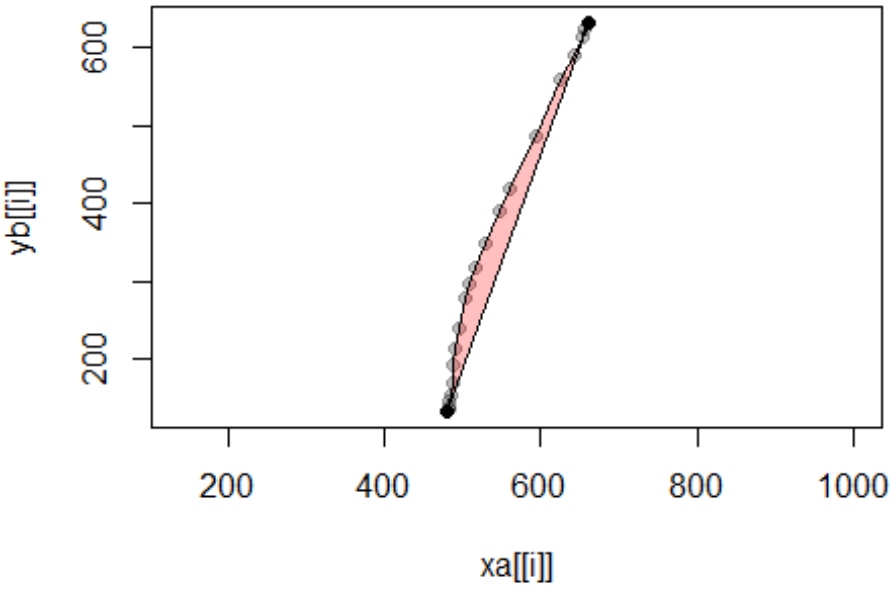

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

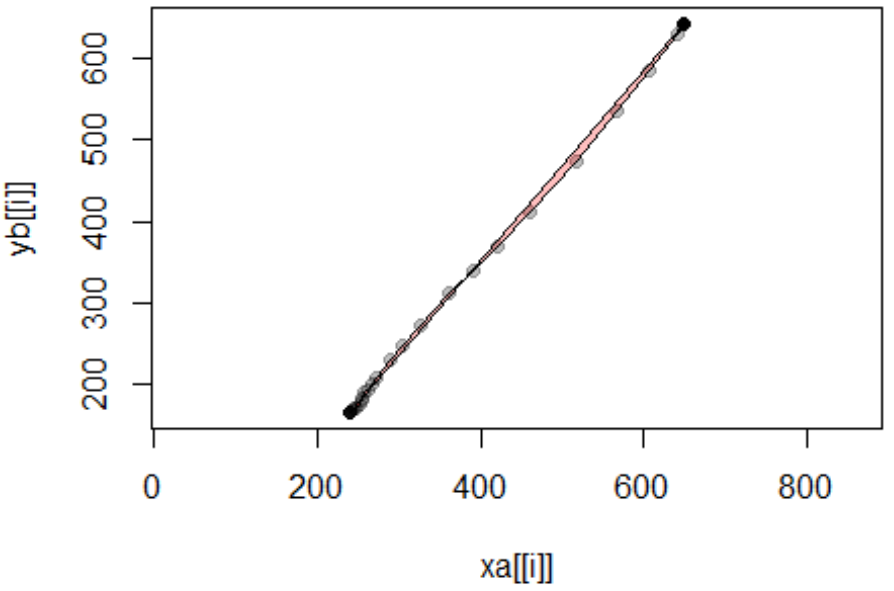

yb[[i]]
200
300
400
500
600
600
800
1000
xa[[i]]

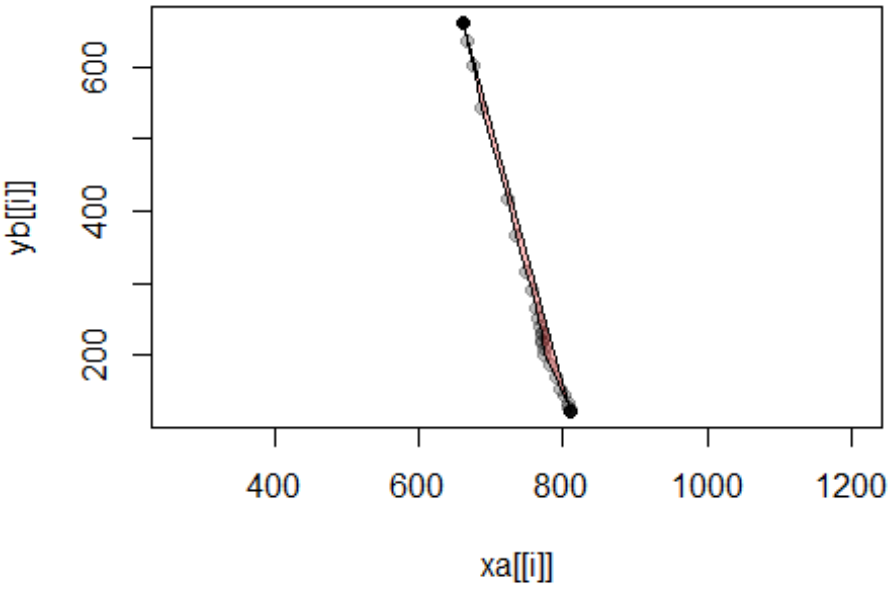

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

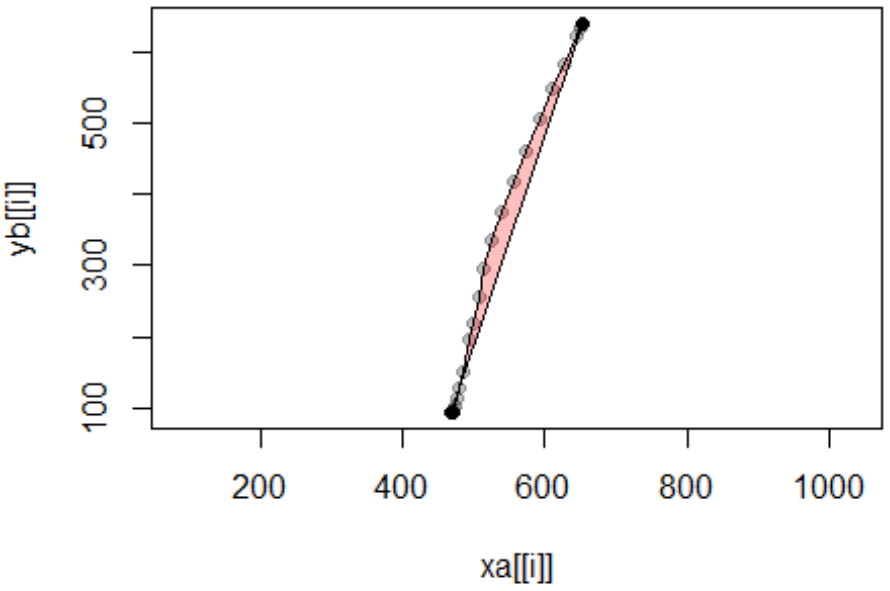

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

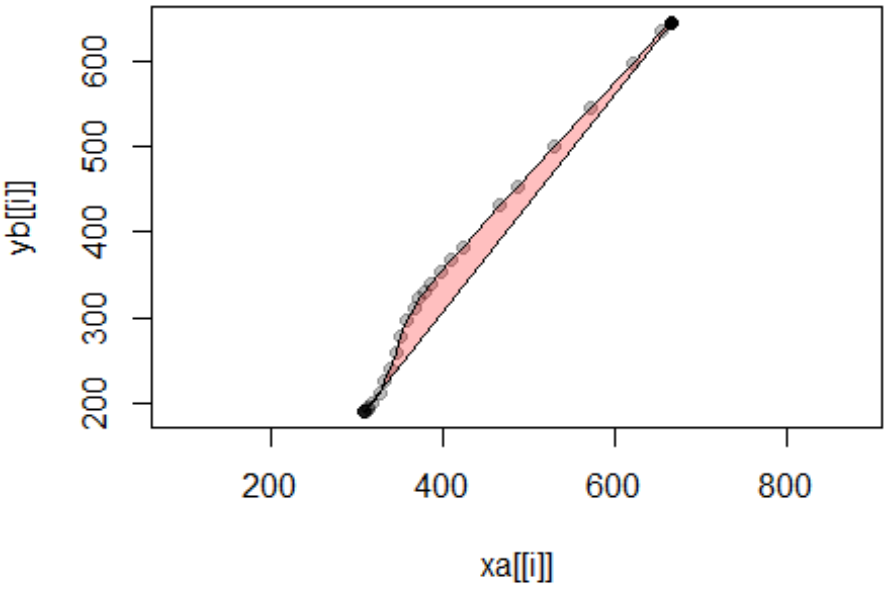

yb[[i]]
200
400
600
0
200
400
600
800
xa[[i]]

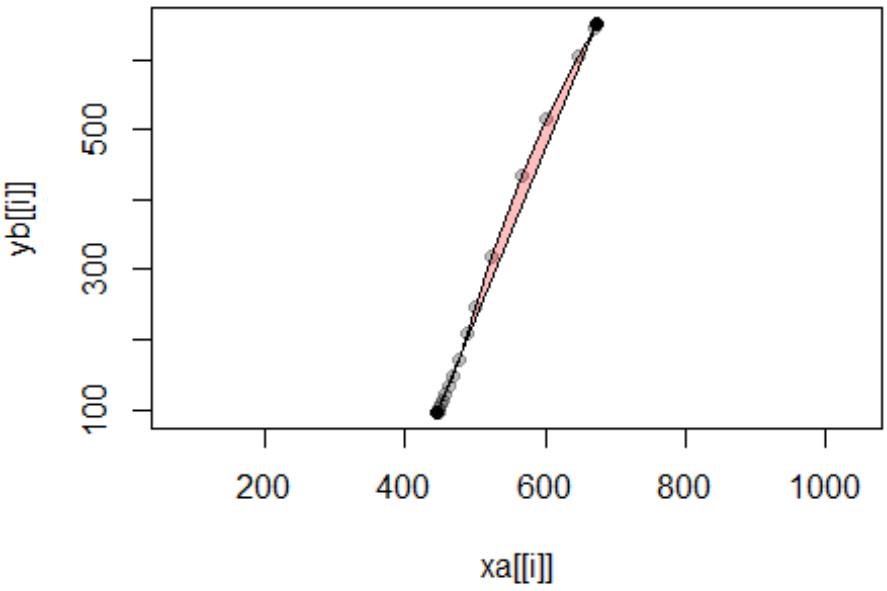

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

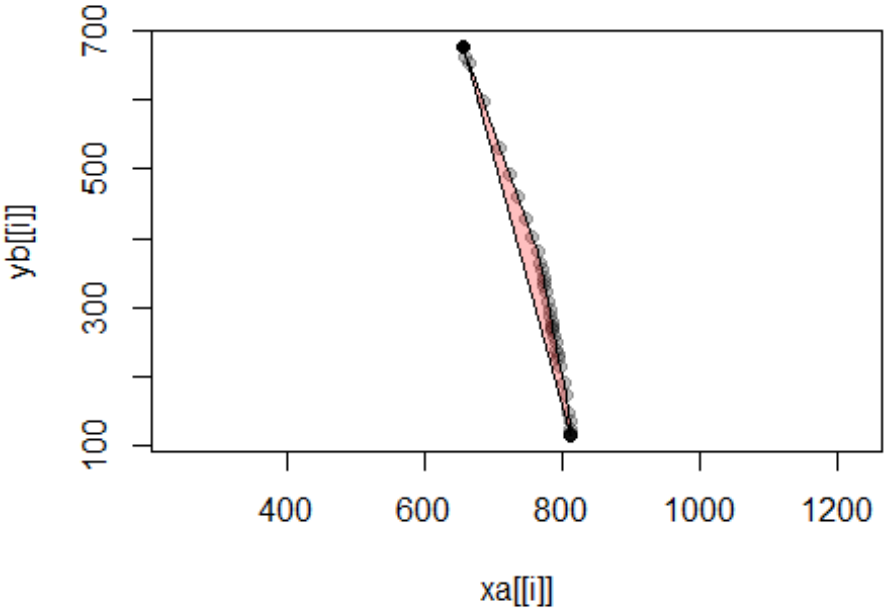

yb[[i]]
200
300
400
500
600
0
200
400
600
800
xa[[i]]

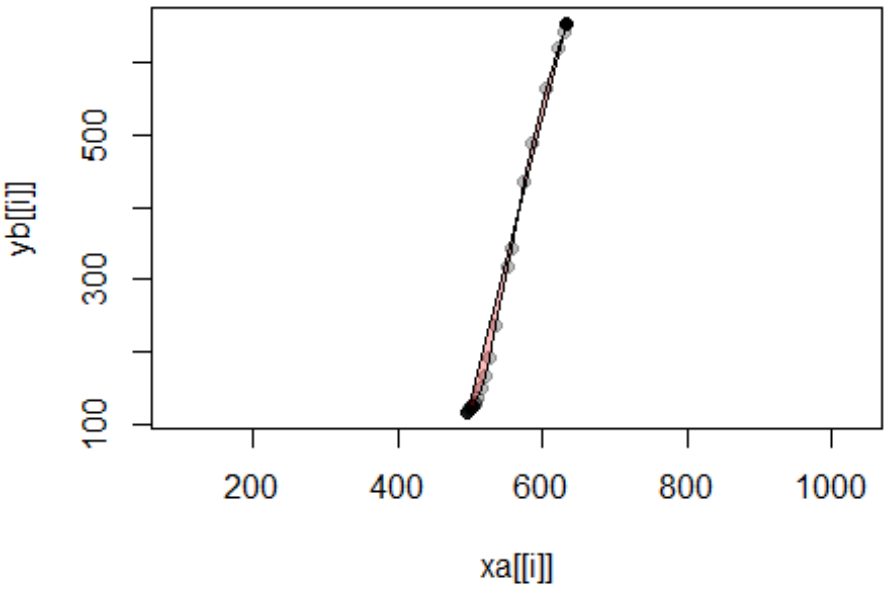

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

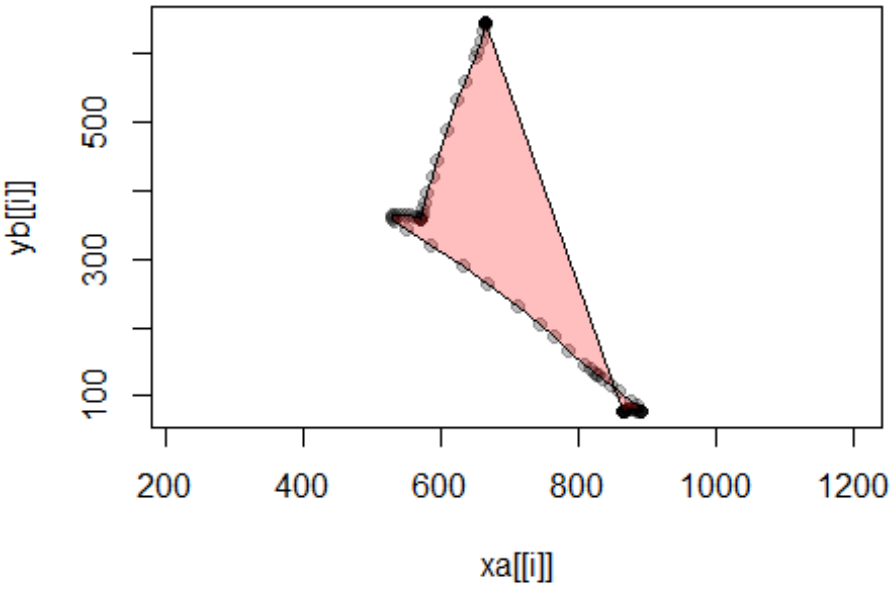

yb[[i]]
500
300
100
200
400
600
800
1000
xa[[i]]

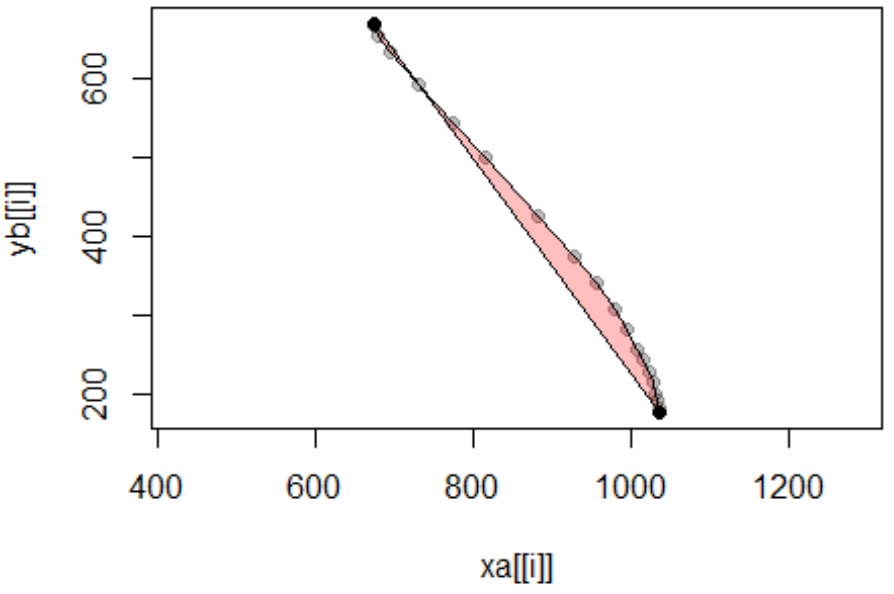

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

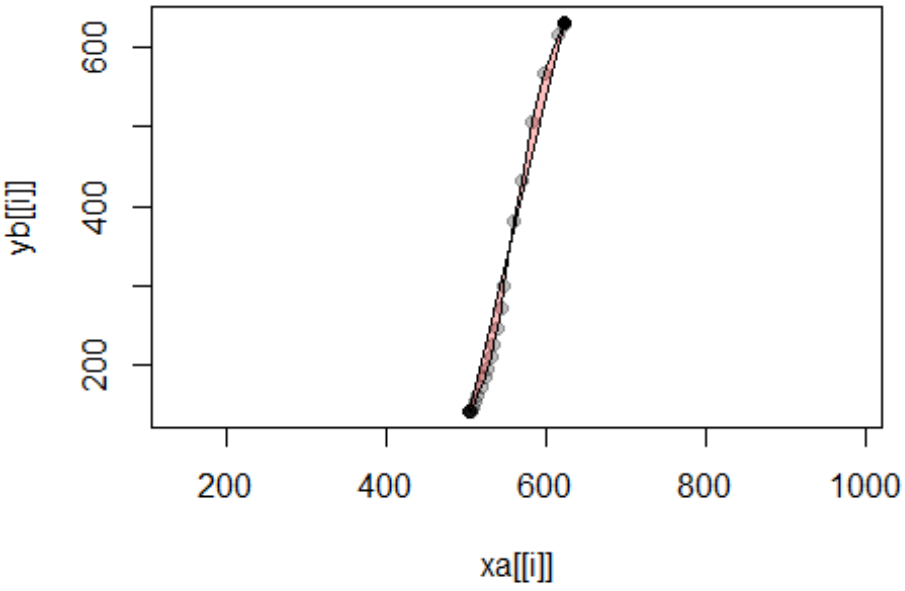

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

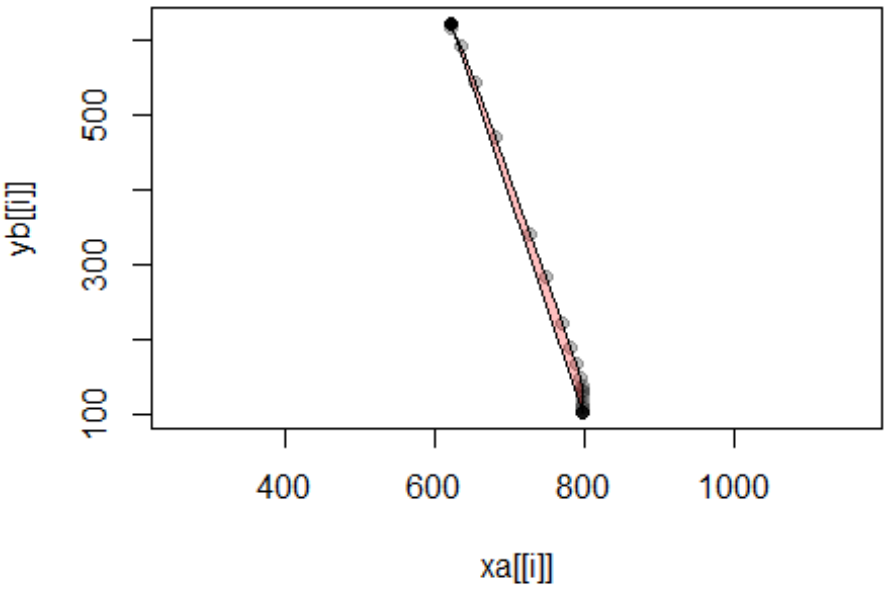

yb[[i]]
100
300
500
700
400
600
800
1000
1200
xa[[i]]

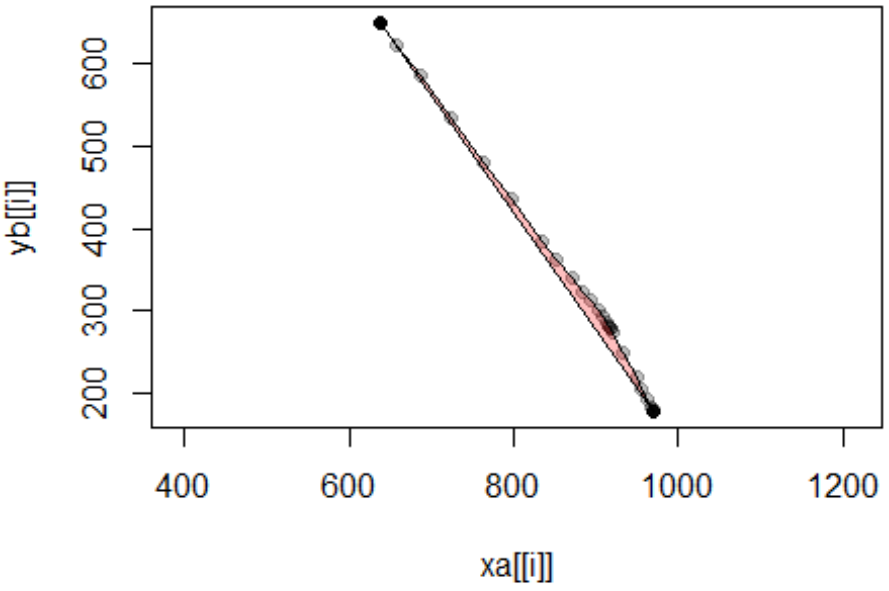

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

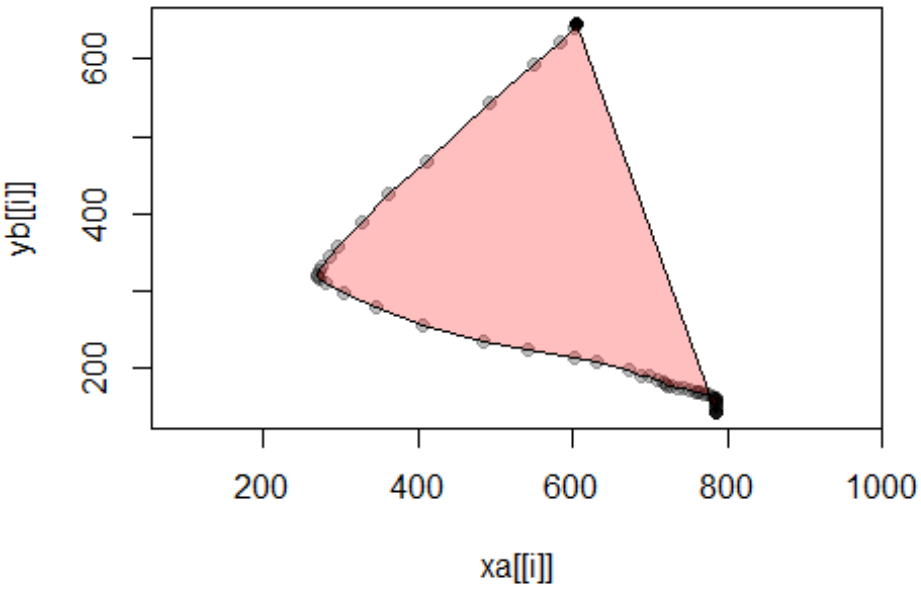

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

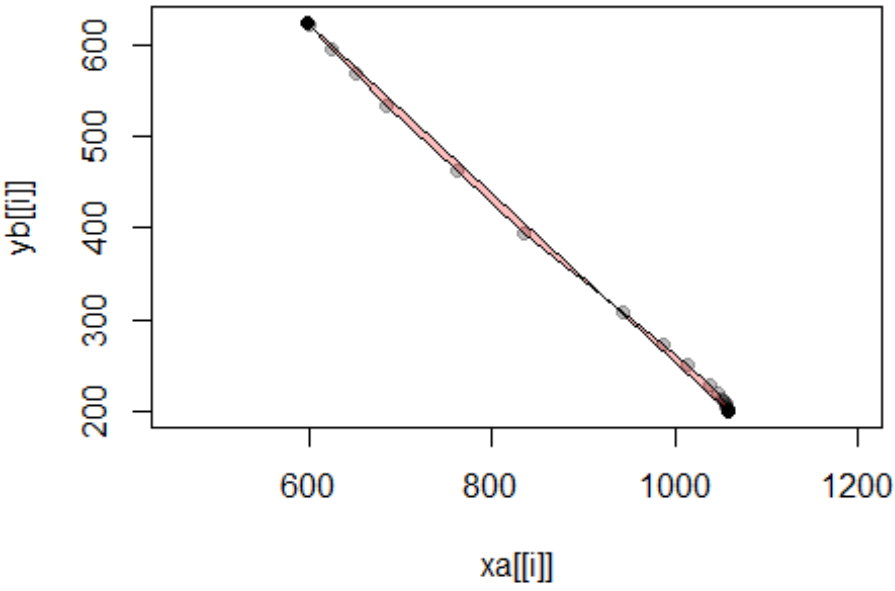

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

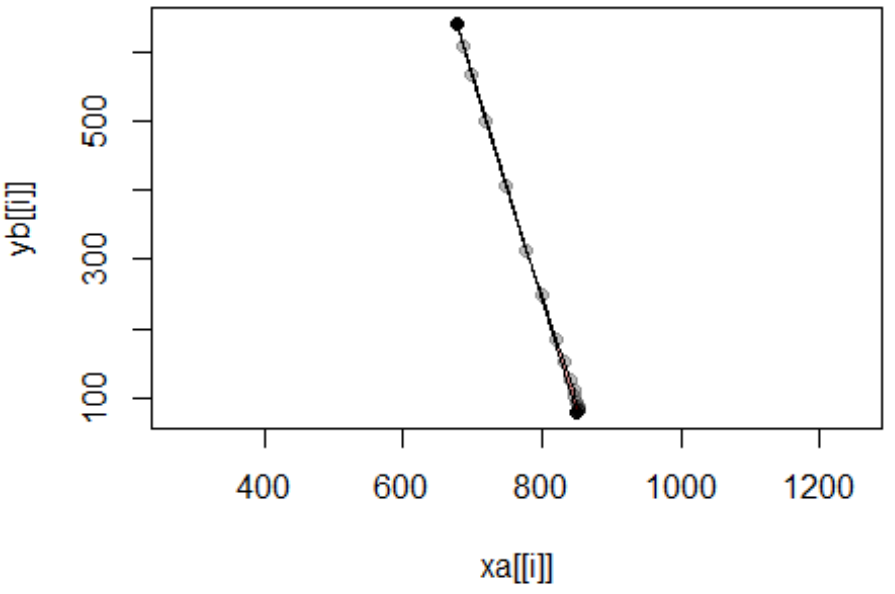

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

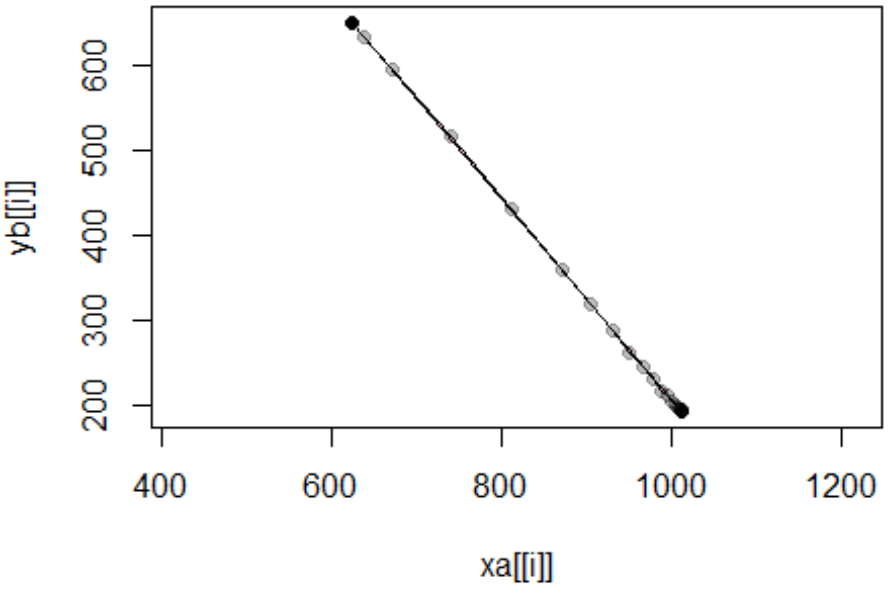

yb[[i]]
100
300
500
400
600
800
1000
xa[[i]]

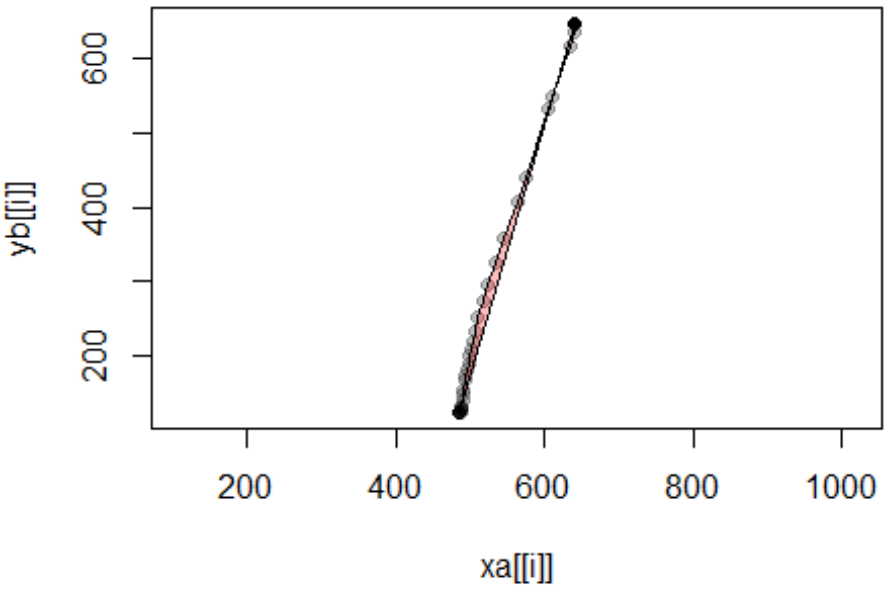

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

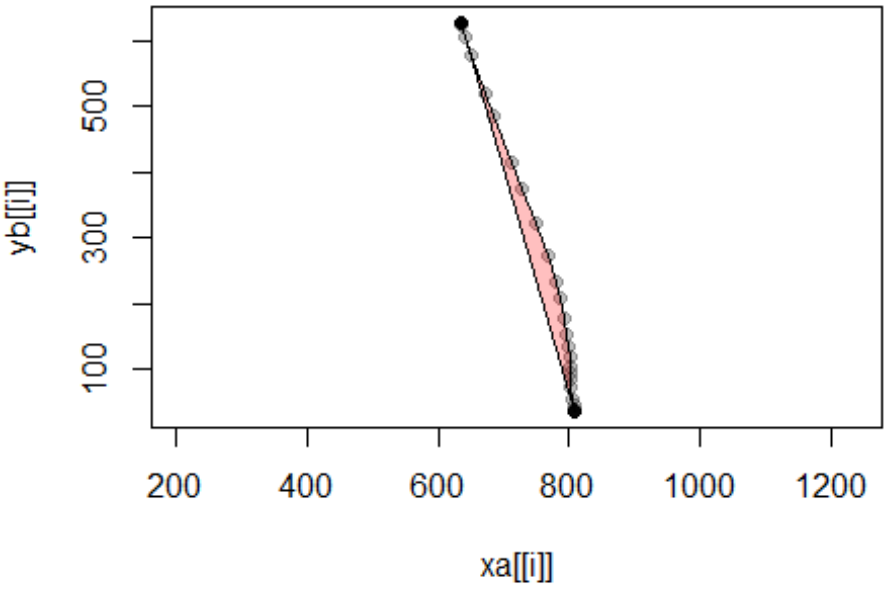

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

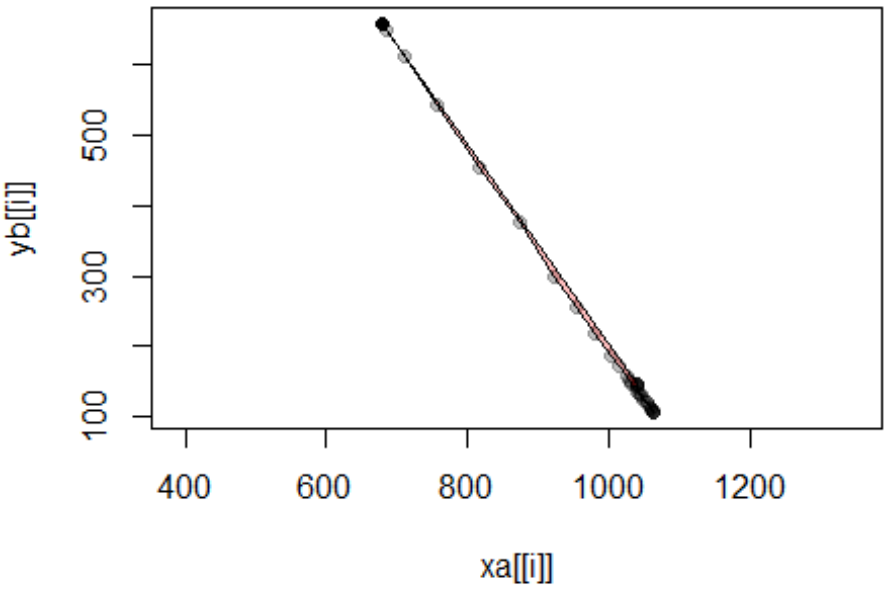

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

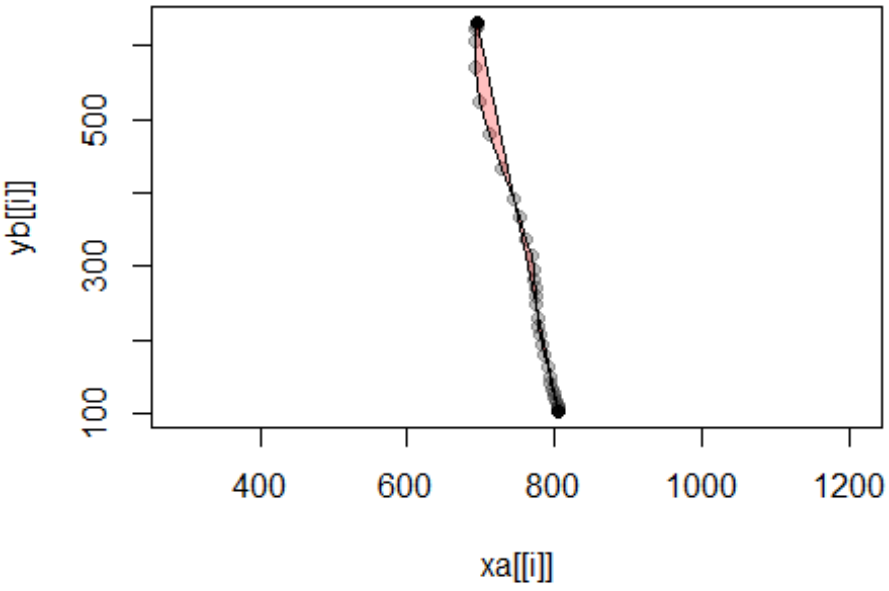

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

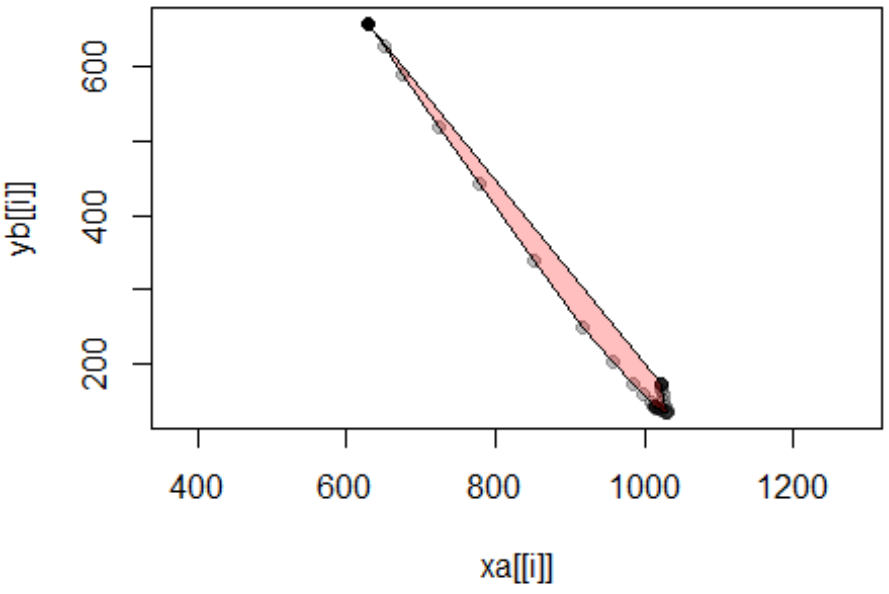

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

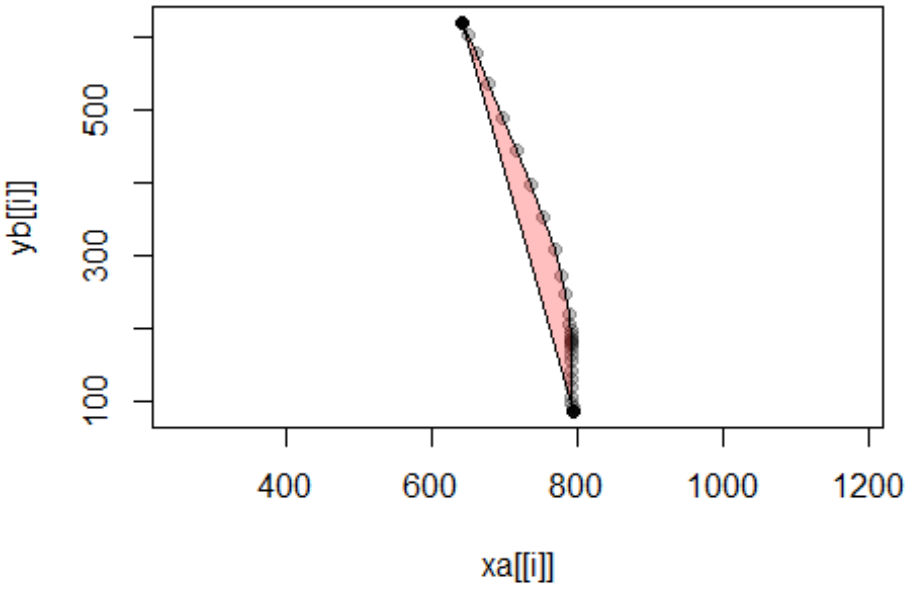

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

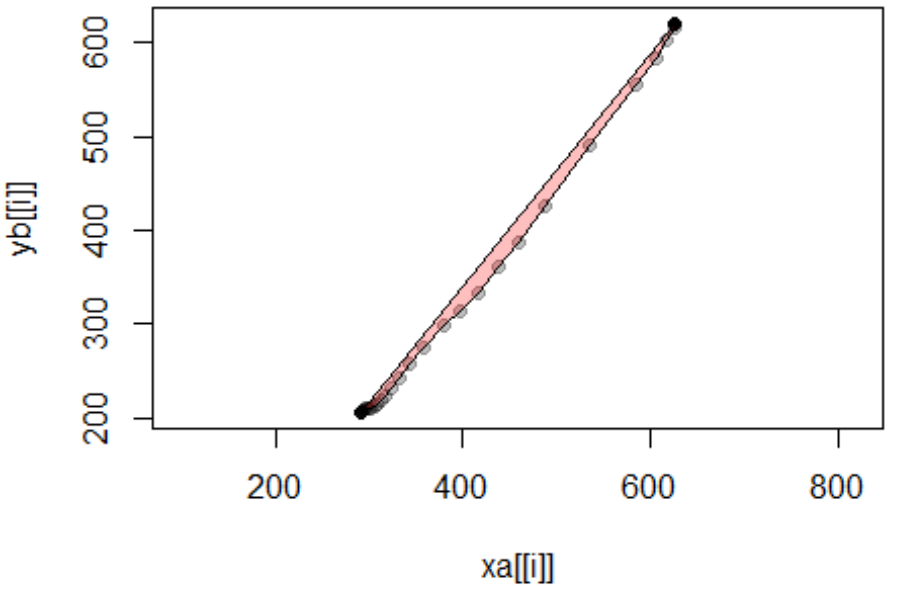

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

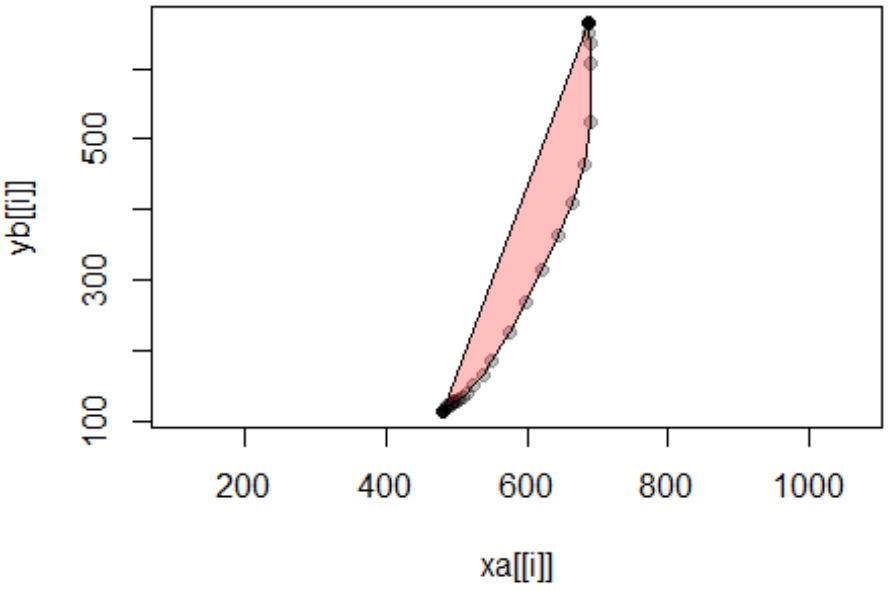

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

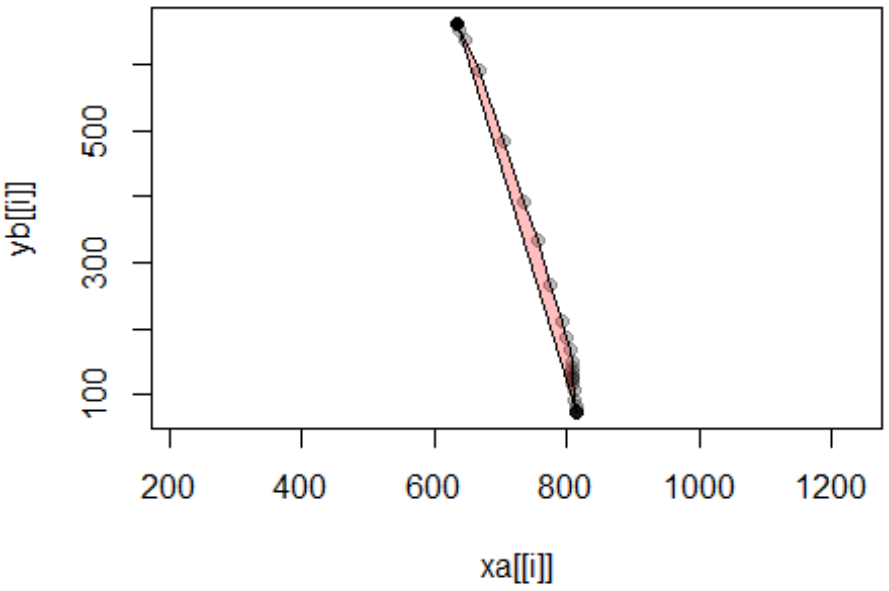

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

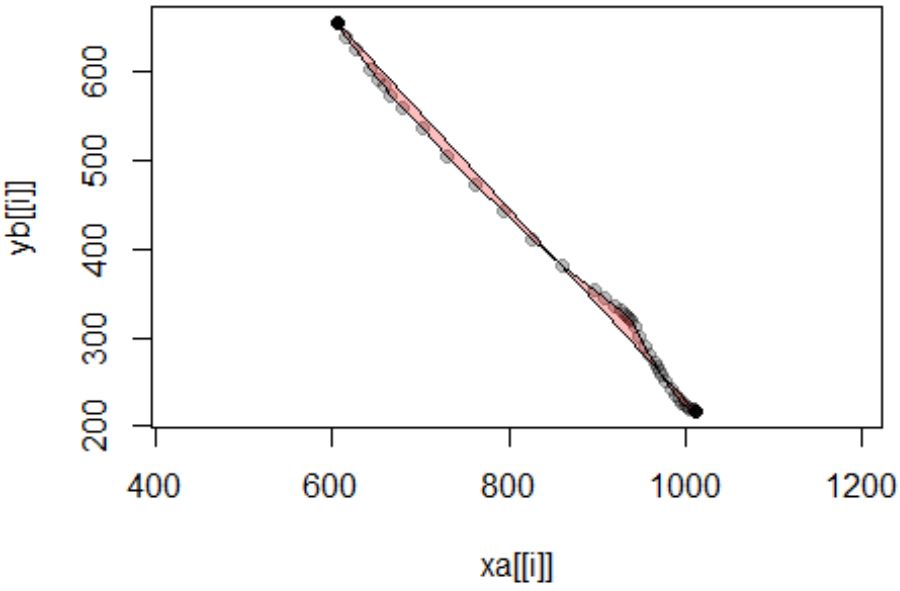

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

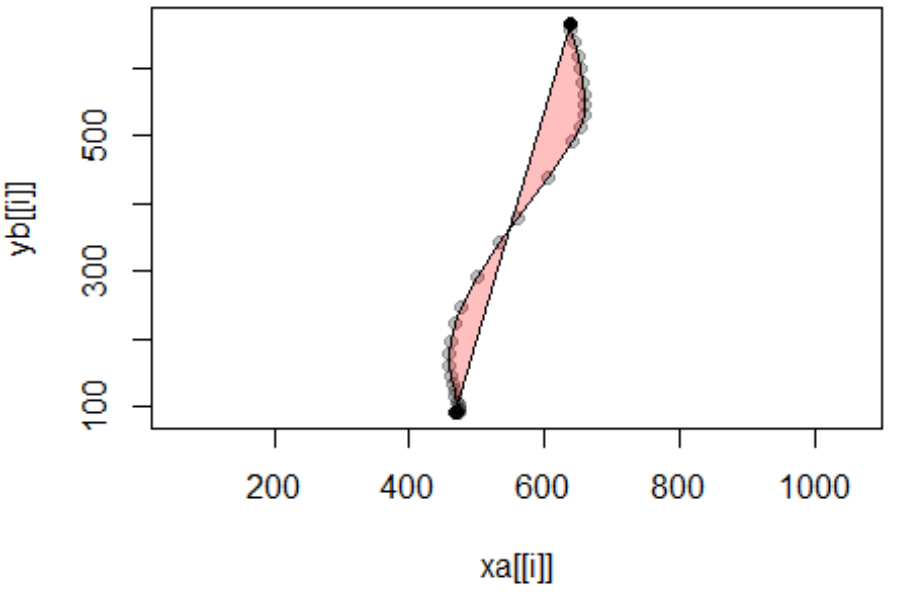

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

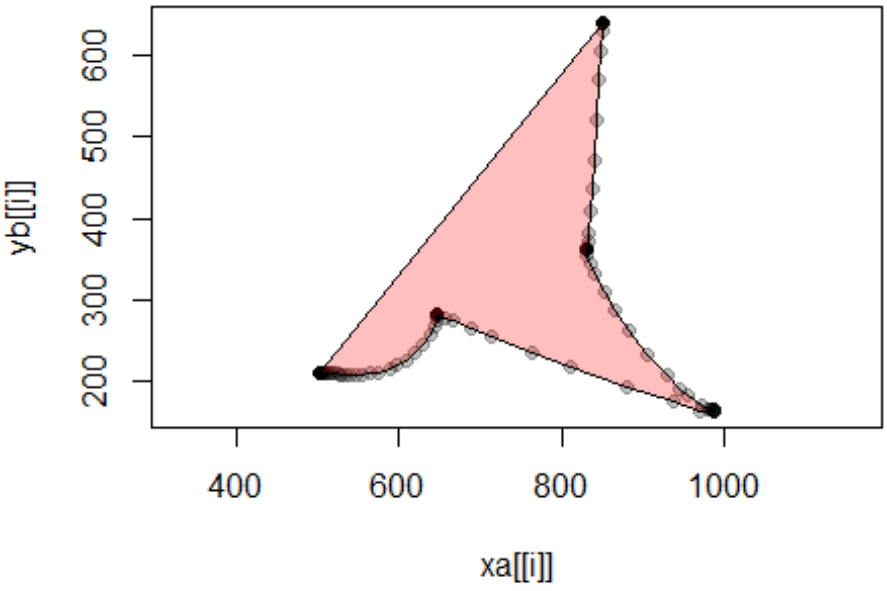

yb[[i]]
200
300
400
500
600
200
400
600
800
xa[[i]]

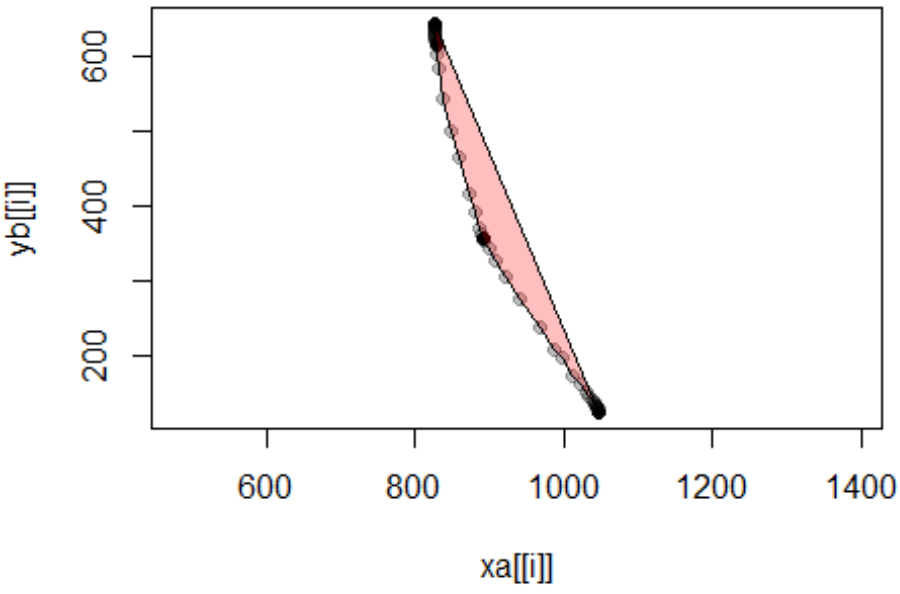

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

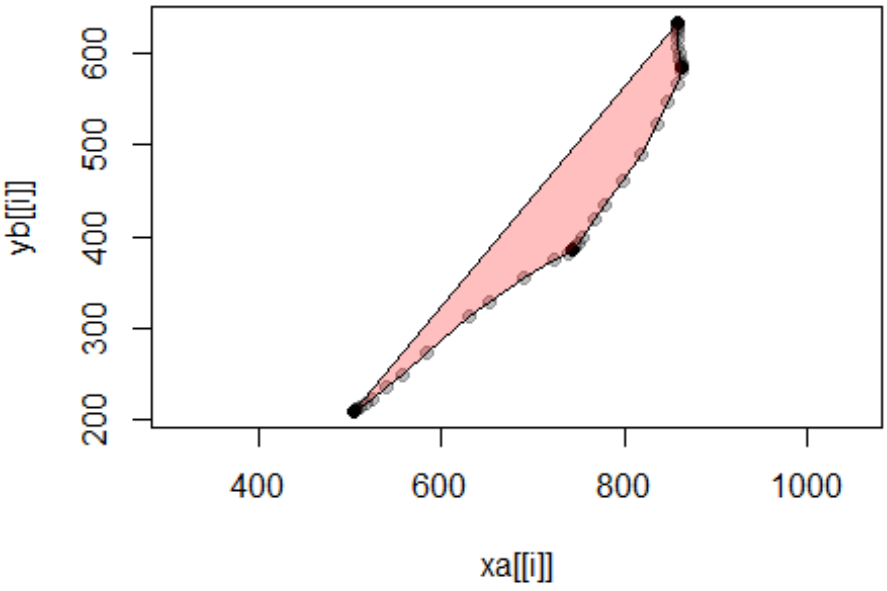

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

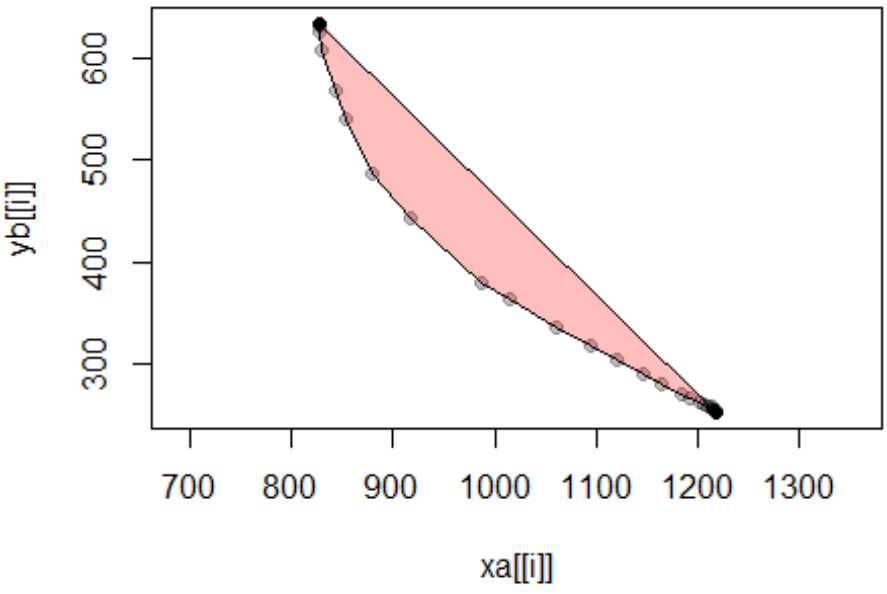

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

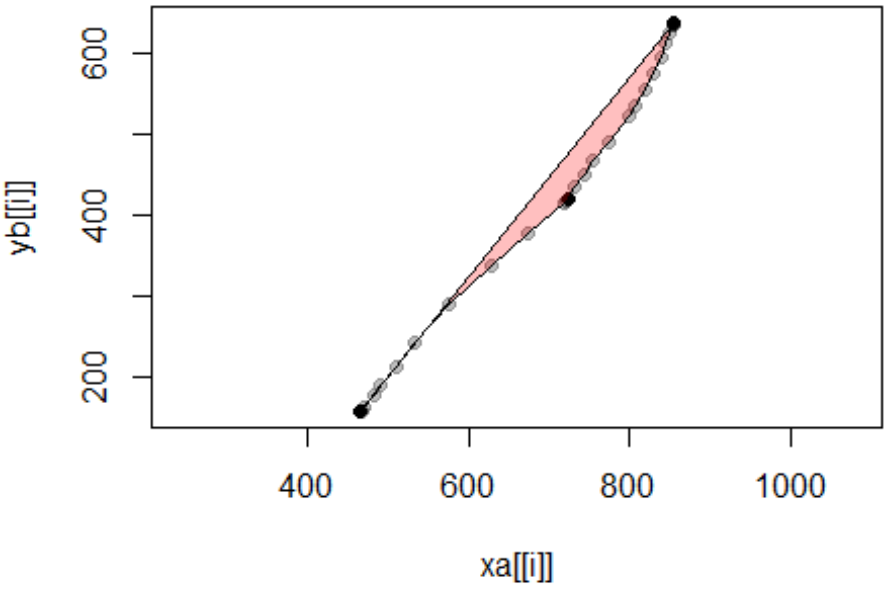

yb[[i]]
100
300
500
200
400
600
800
1000
xa[[i]]

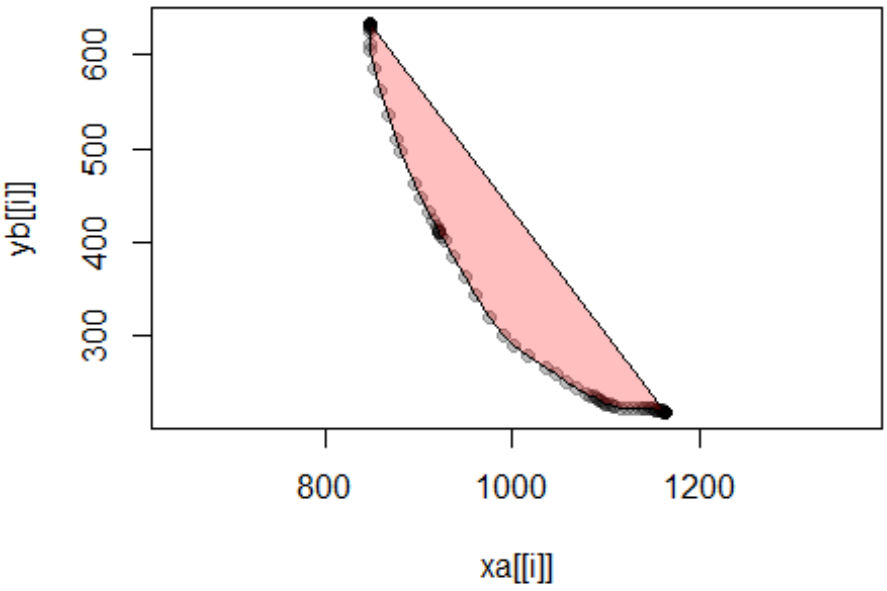

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

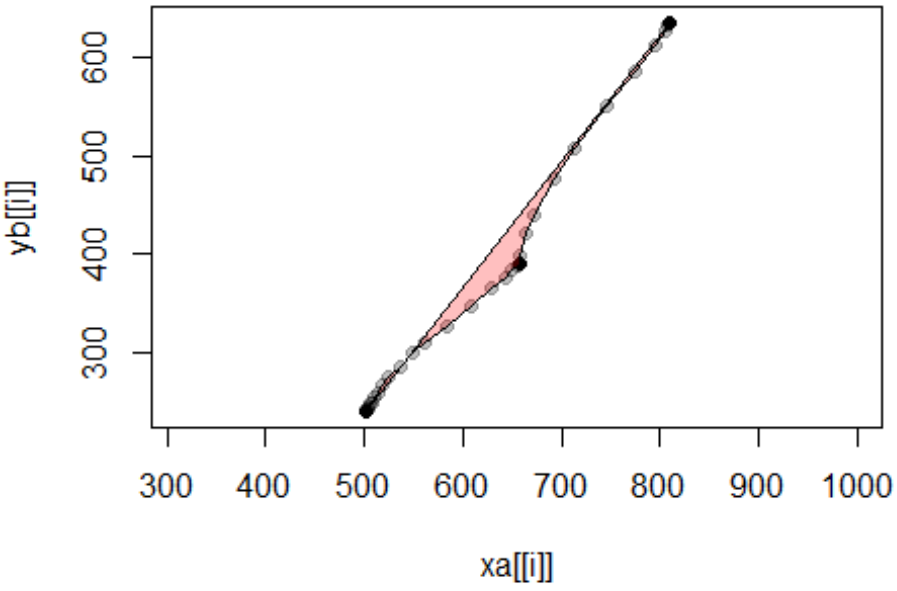

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

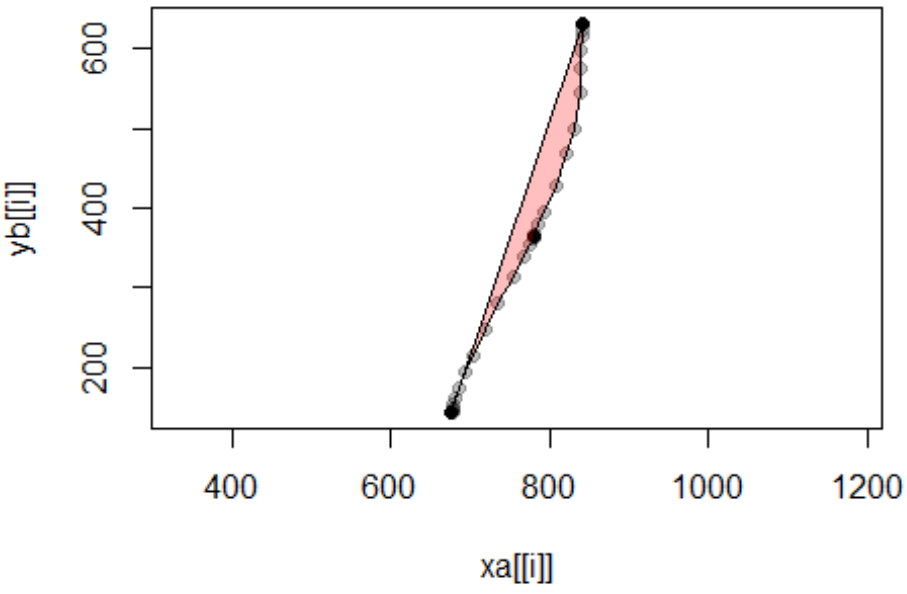

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

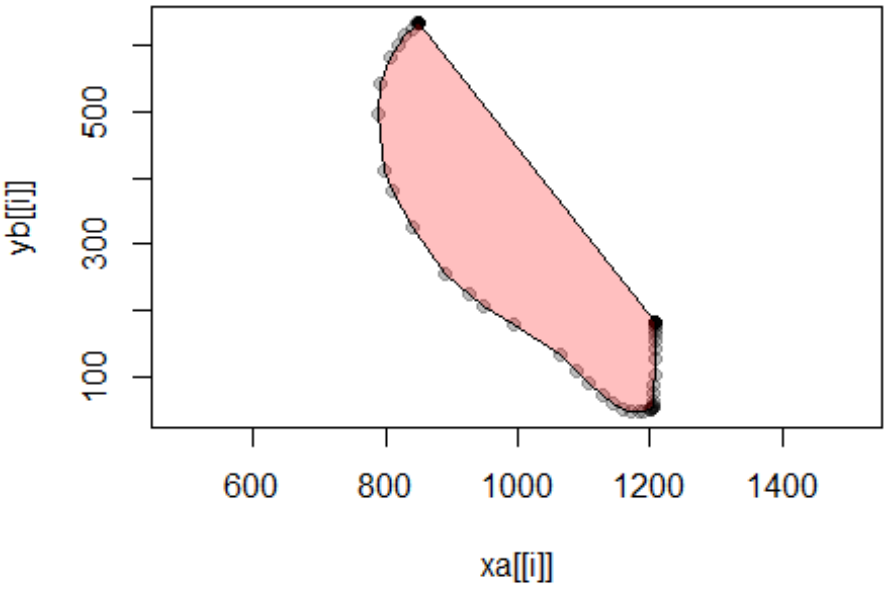

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

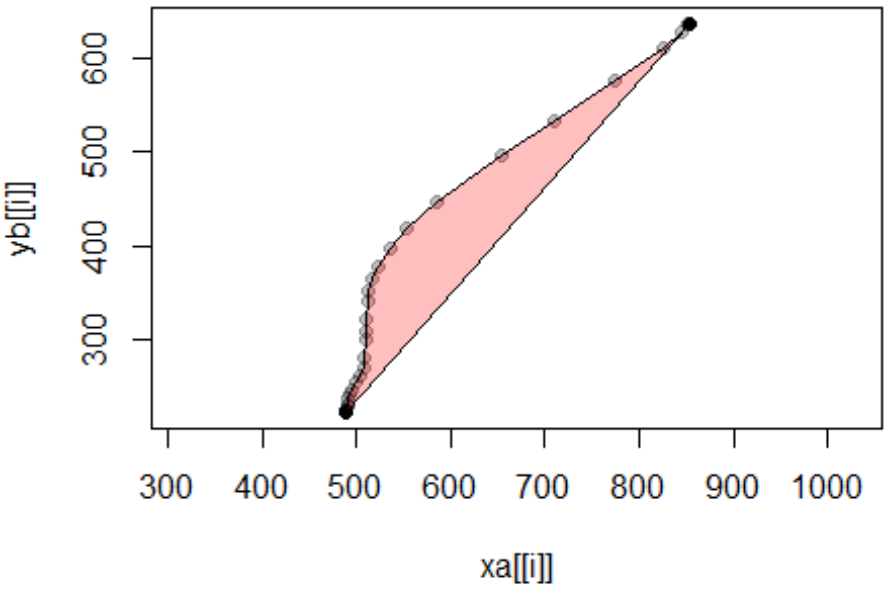

yb[[i]]
200
400
600
400
600
800
1000
xa[[i]]

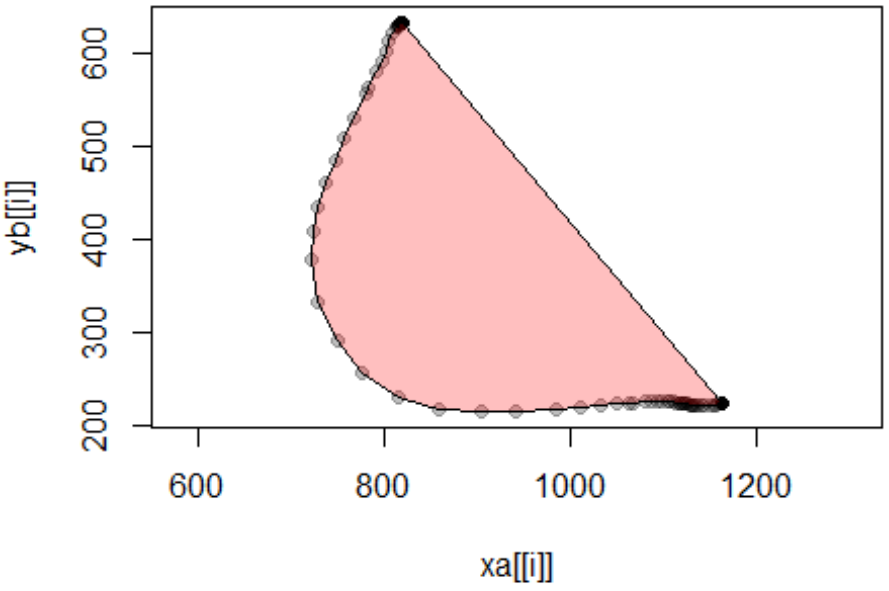

yb[[i]]
300
400
500
600
800
1000
1200
xa[[i]]

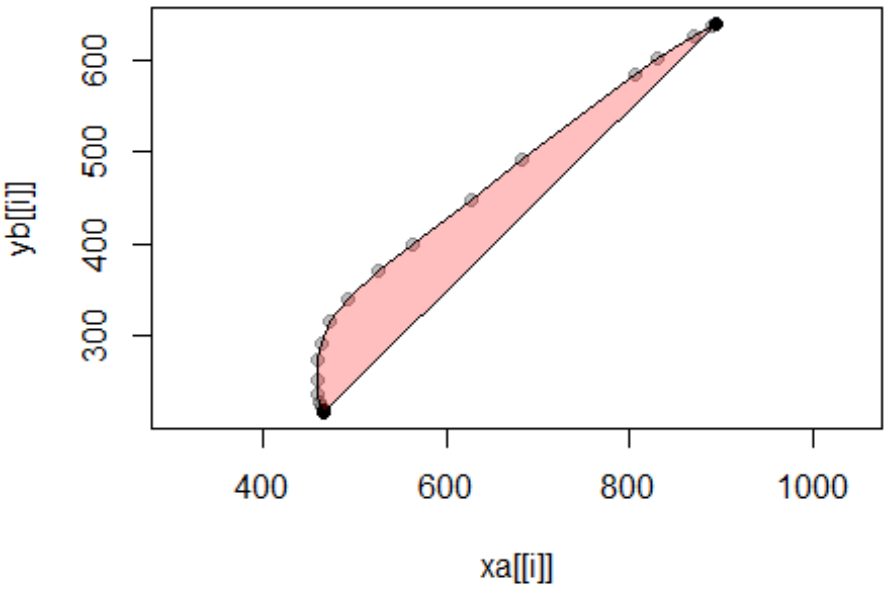

yb[[i]]
300
400
500
600
xa[[i]]
300
400
500
600
700
800
900
1000

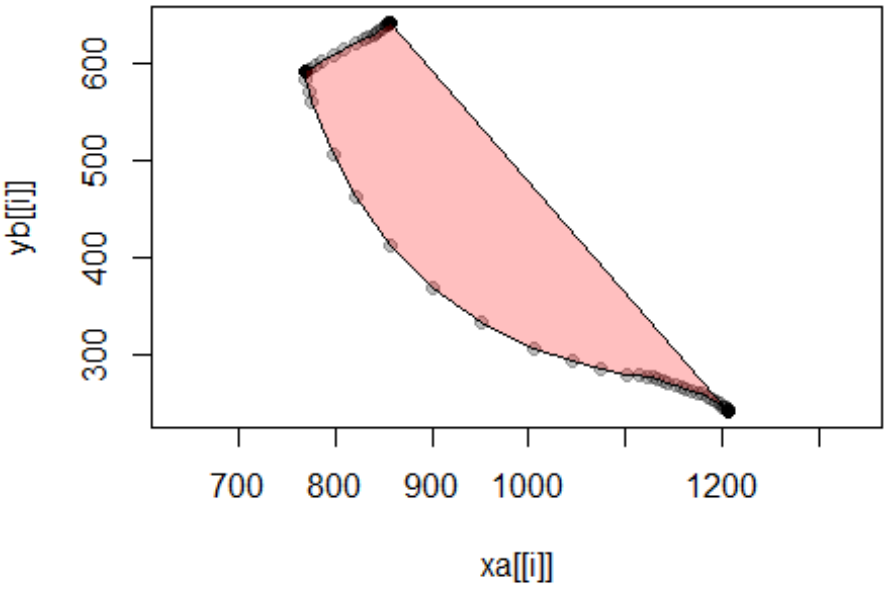

yb[[i]]
200
400
600
xa[[i]]
400
600
800
1000
1200

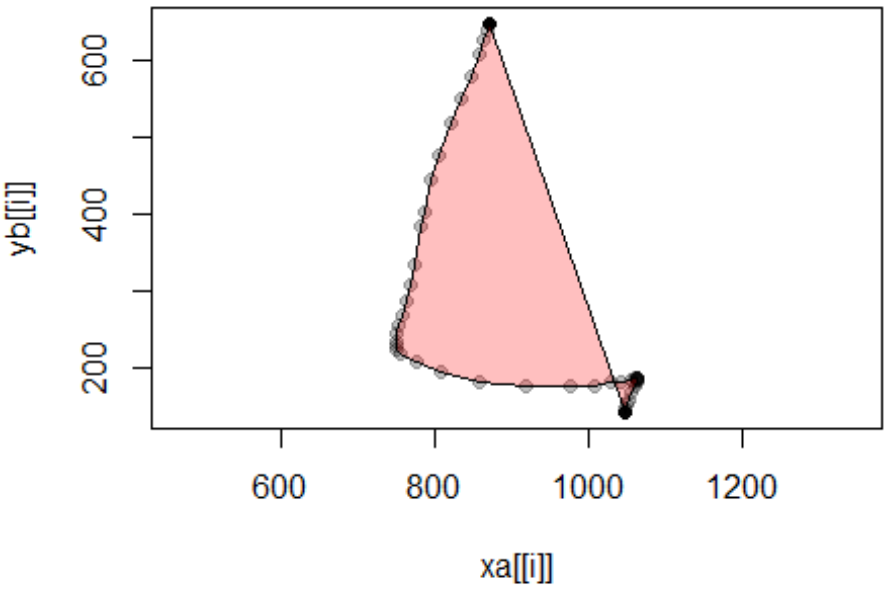

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

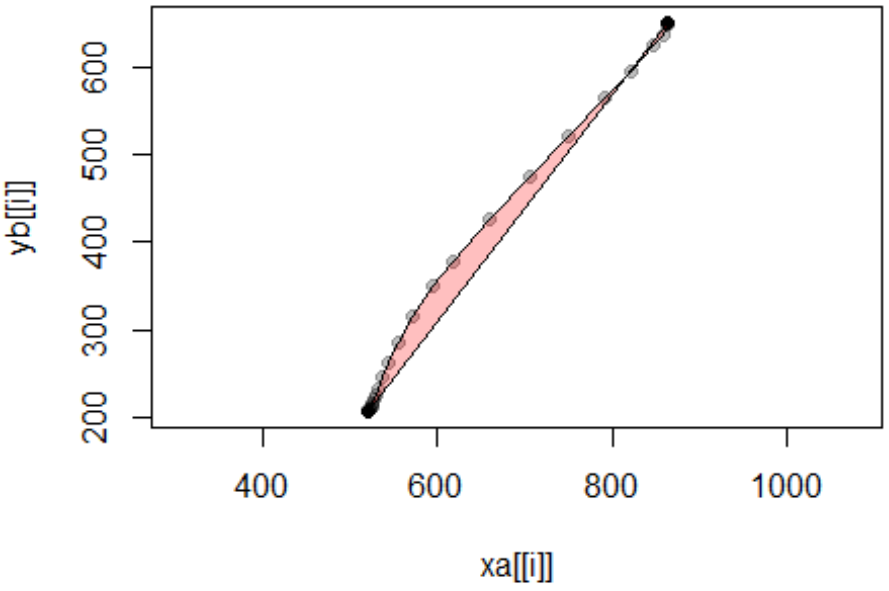

yb[[i]]
300
400
500
600
300
400
500
600
700
800
900
1000
xa[[i]]

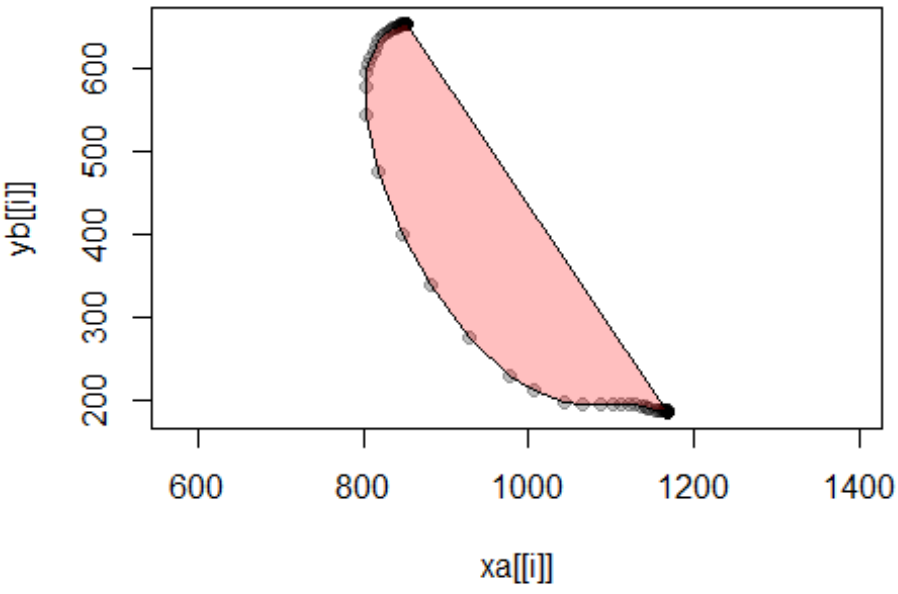

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

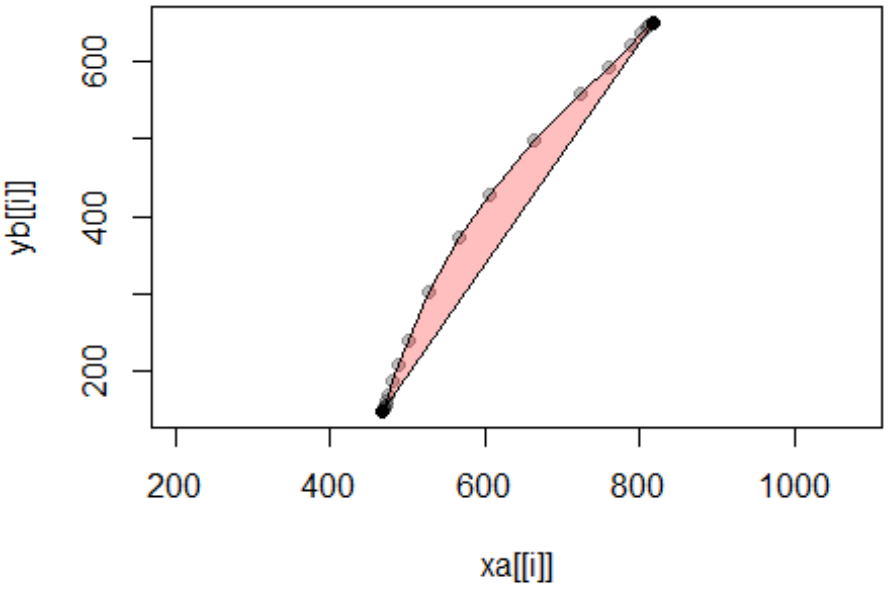

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

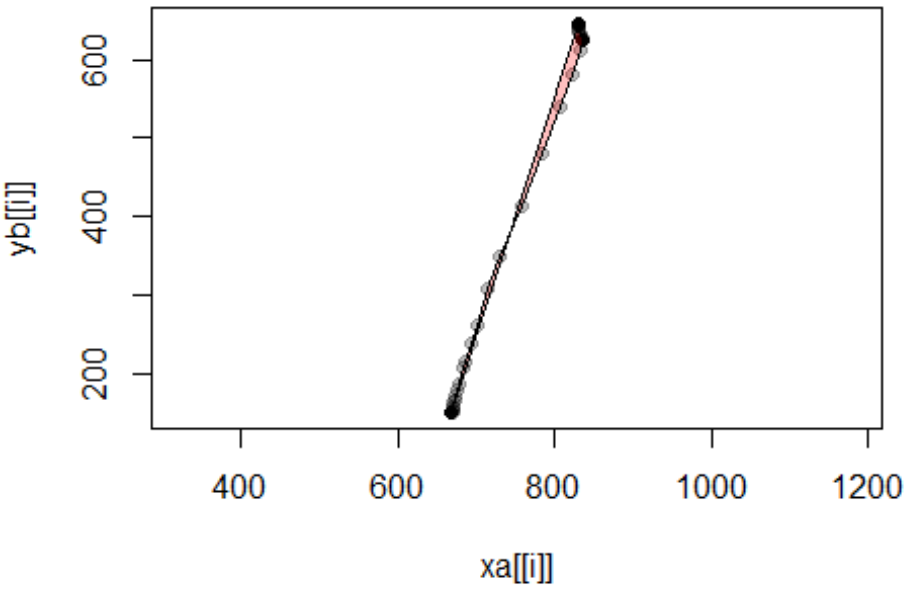

yb[[i]]
300
400
500
600
700
800
900
1000
1200
xa[[i]]

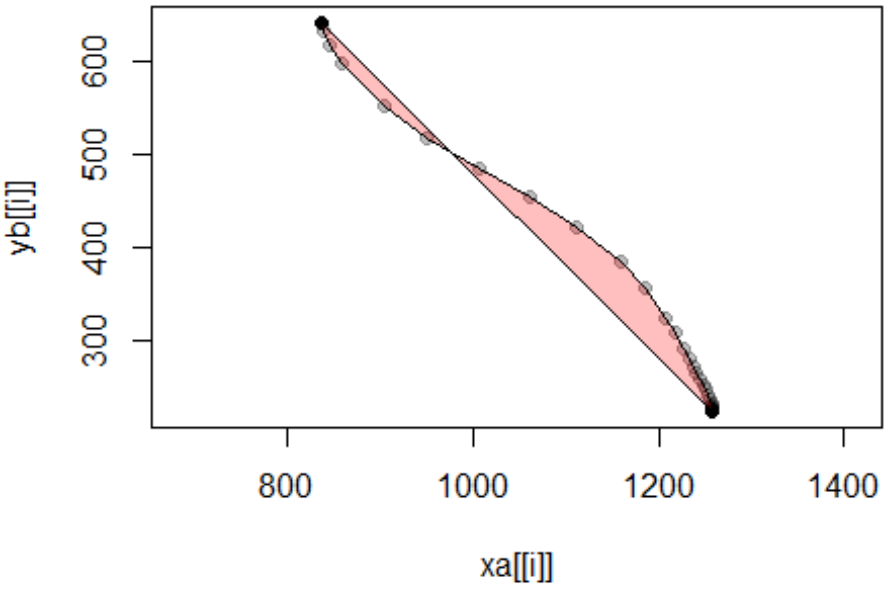

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

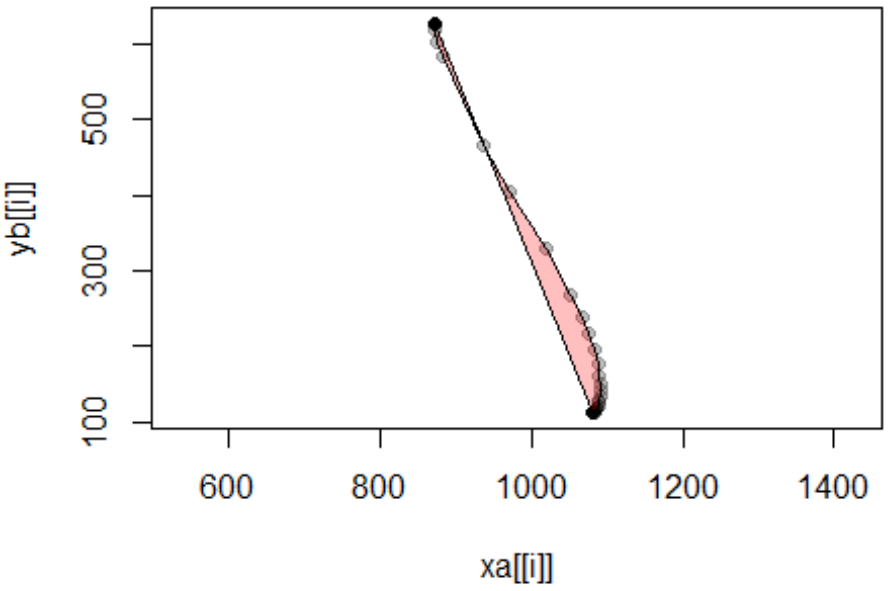

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

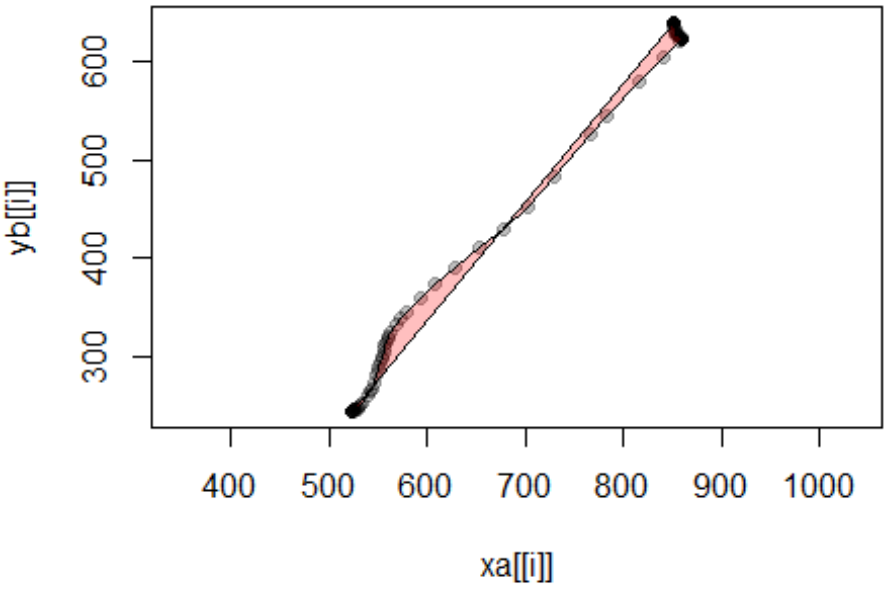

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

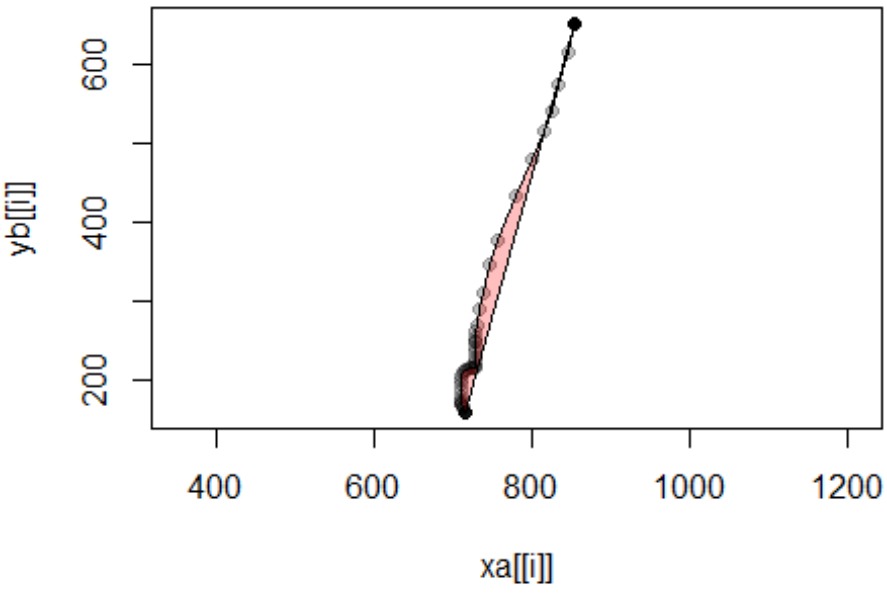

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

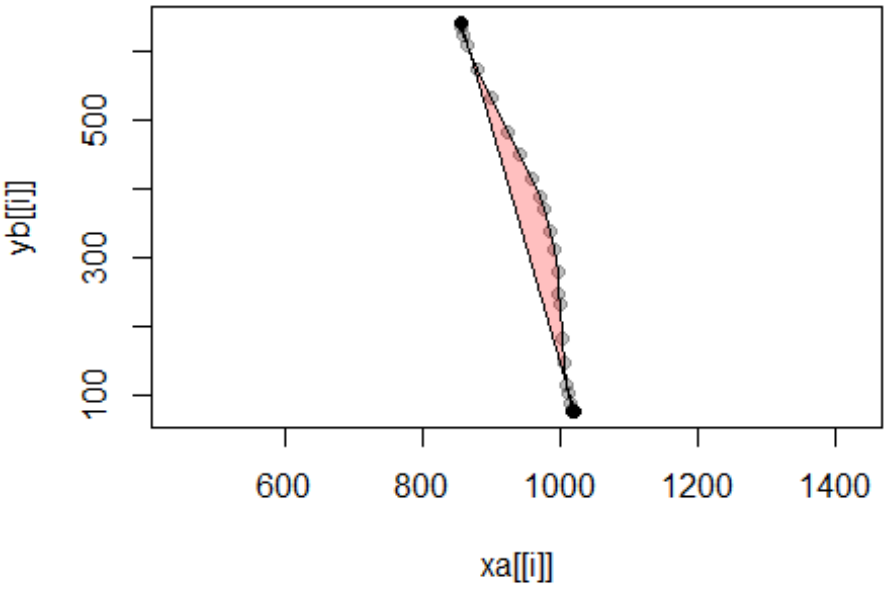

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

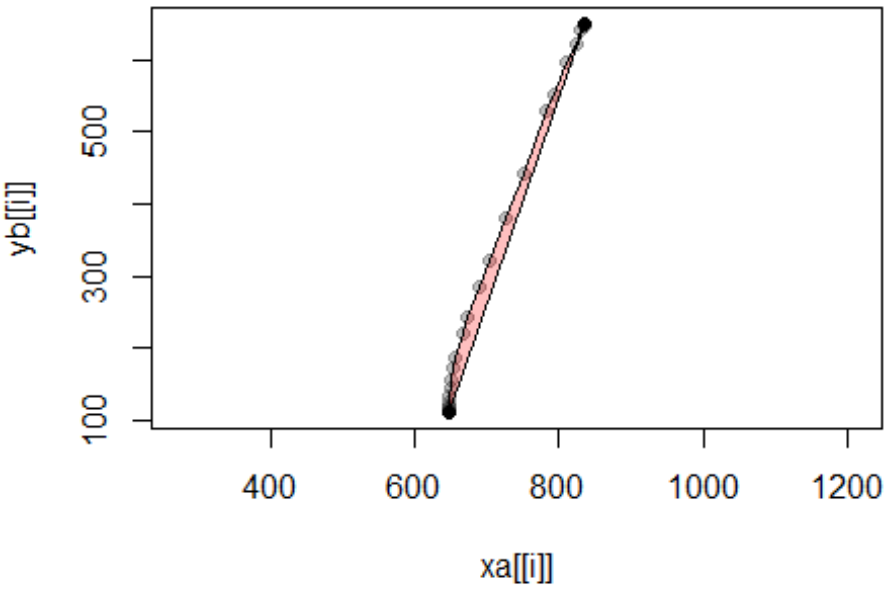

yb[[i]]
300
400
500
600
800
1000
1200
1400
xa[[i]]

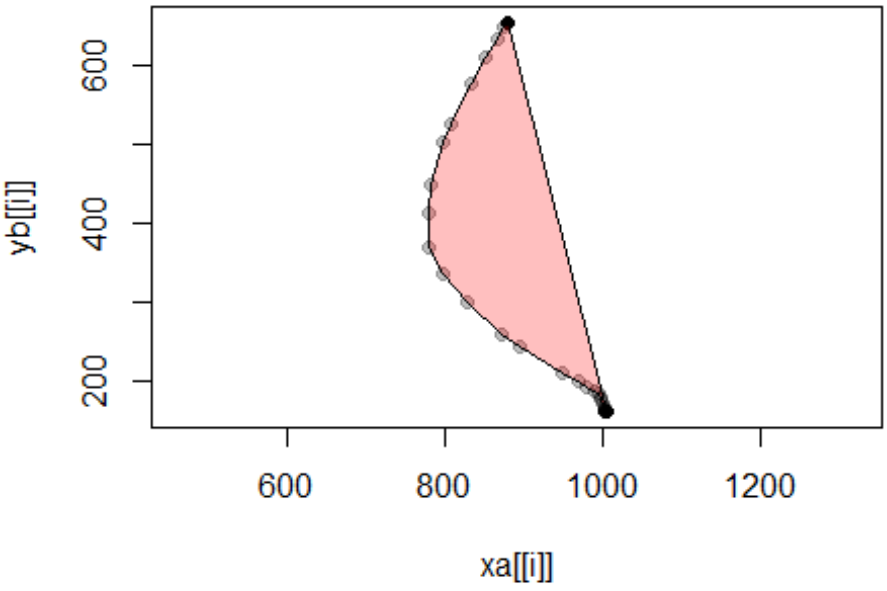

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

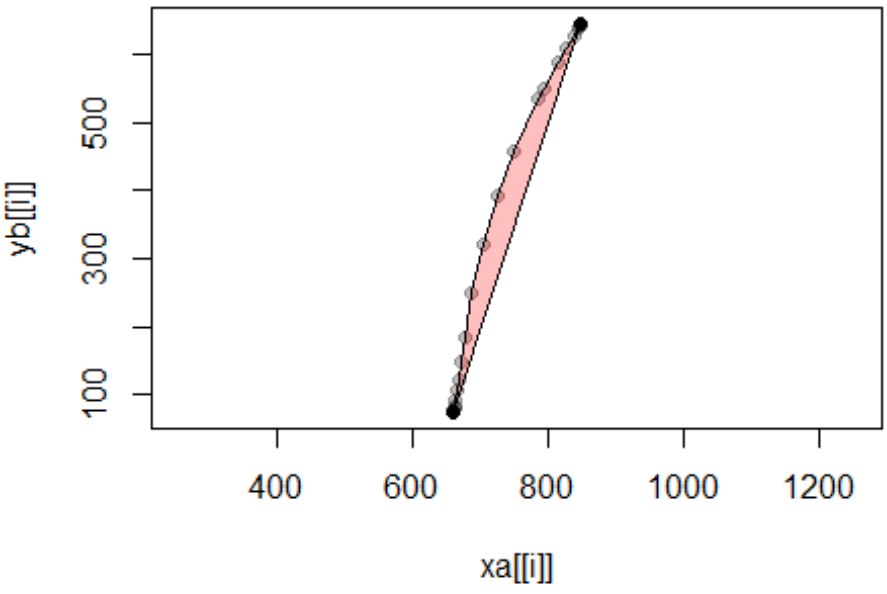

yb[[i]]
300
400
500
600
400
500
600
700
800
900
1000
xa[[i]]

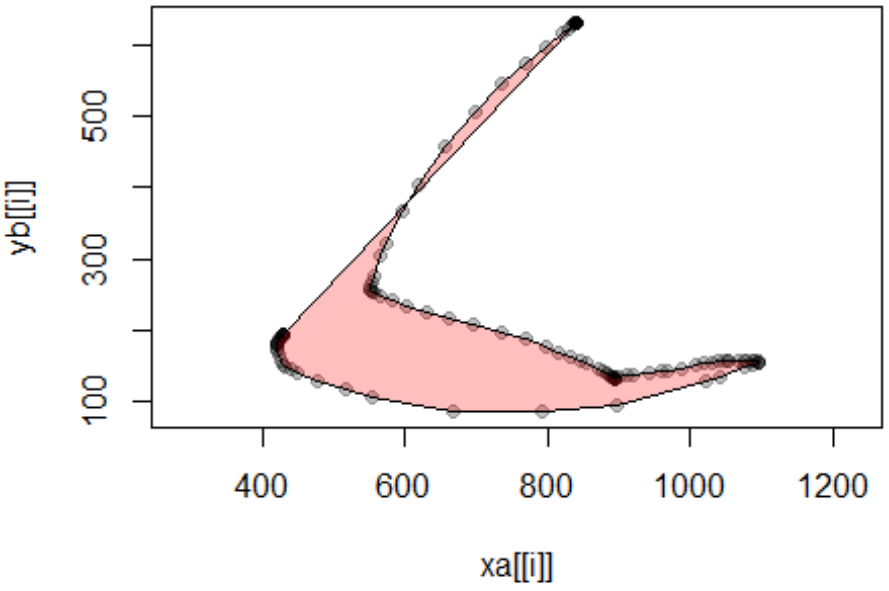

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

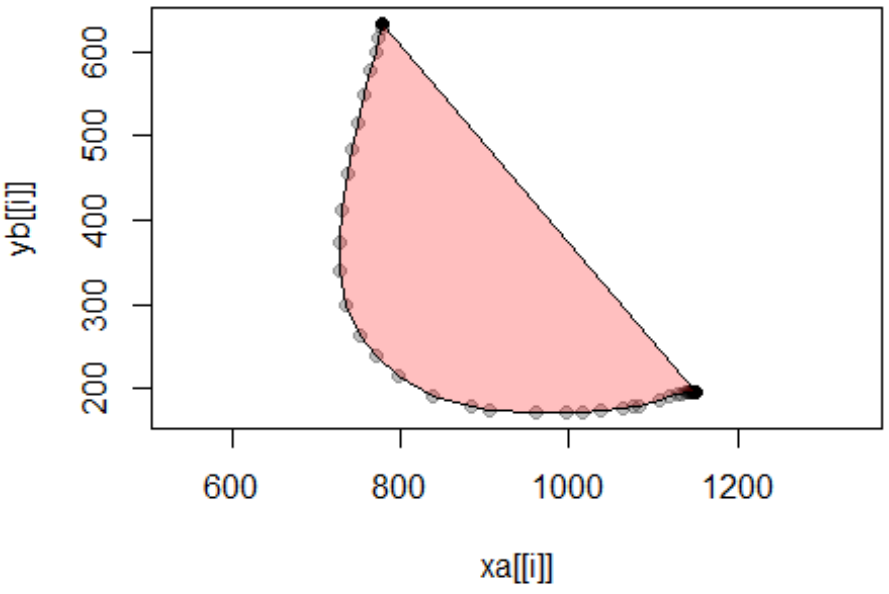

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

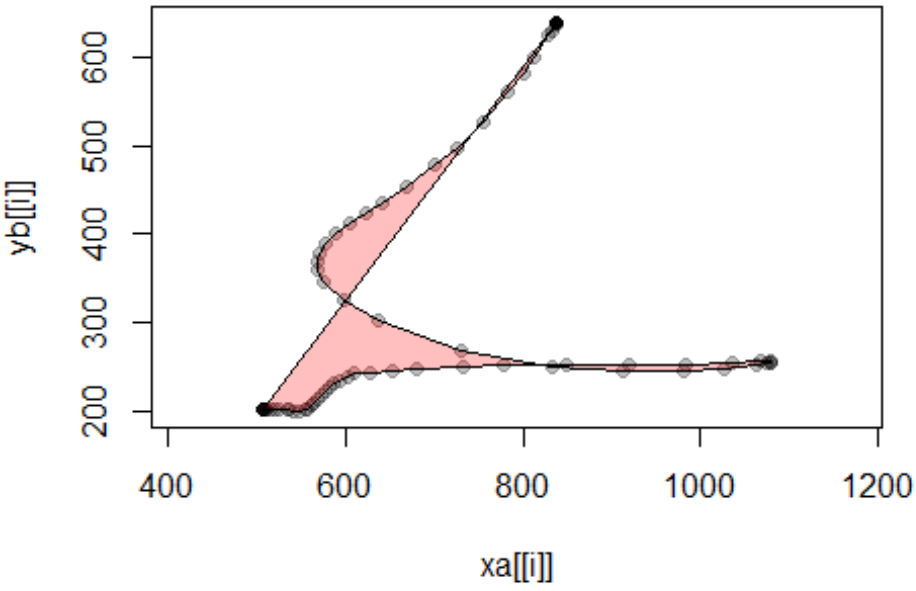

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

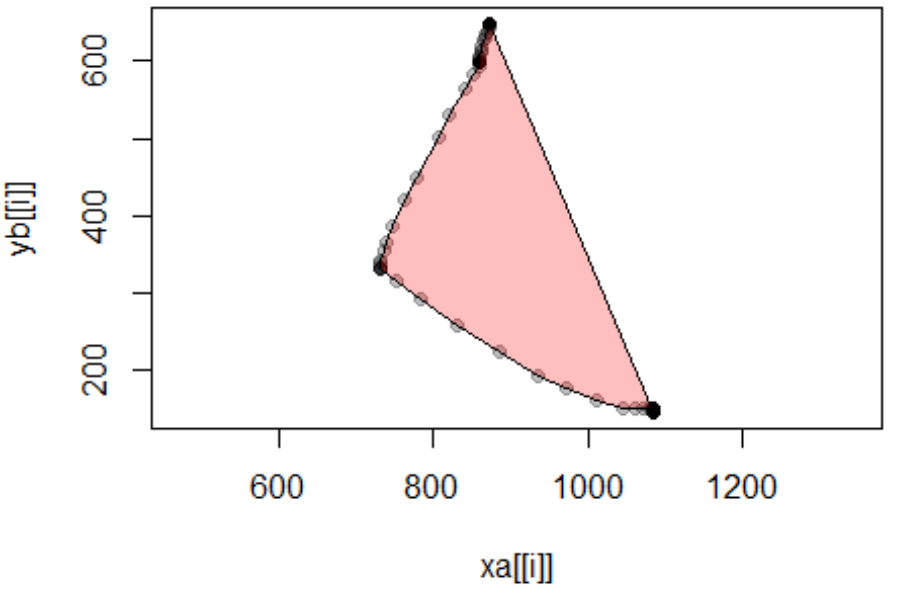

yb[[i]]
600
400
200
600
800
1000
1200
xa[[i]]

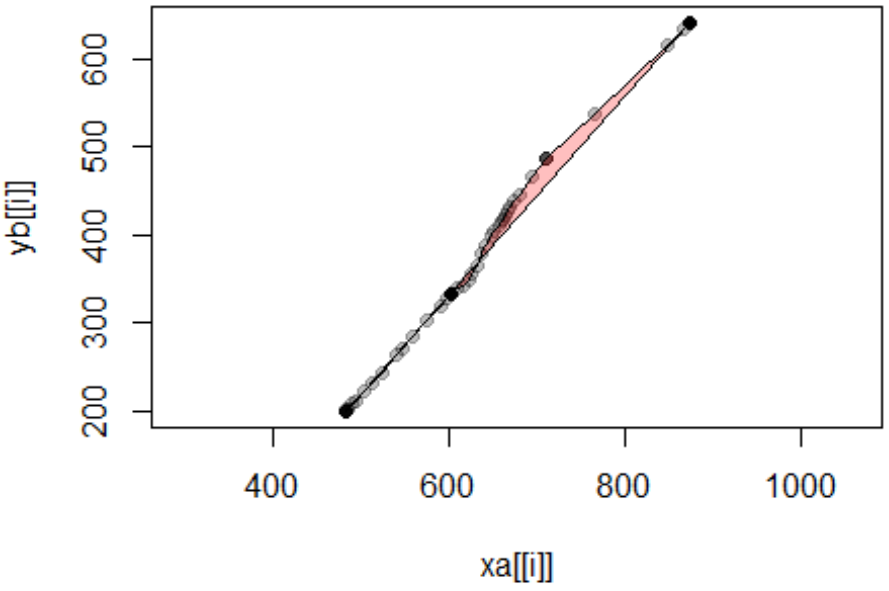

yb[[i]]
500
300
100
400
600
800
1000
1200
xa[[i]]

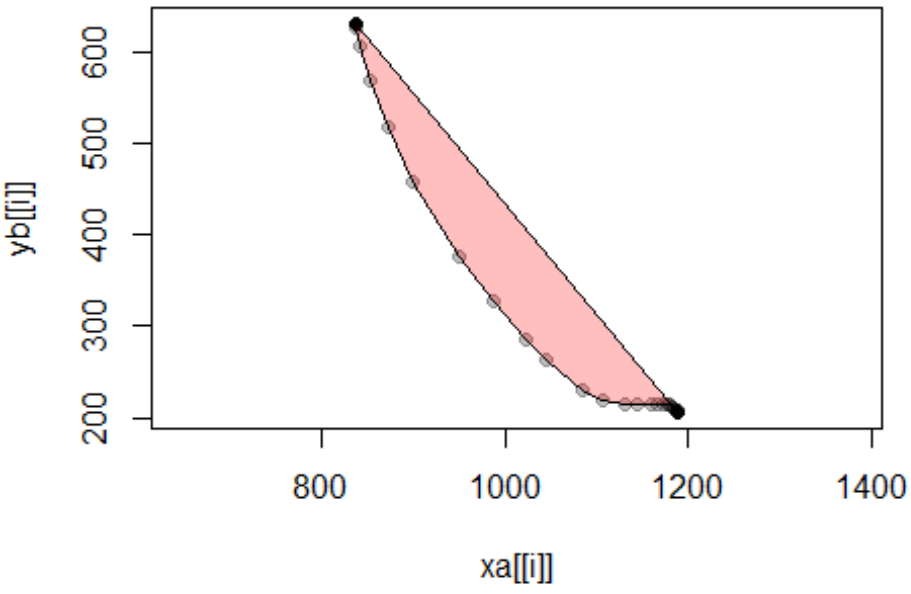

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

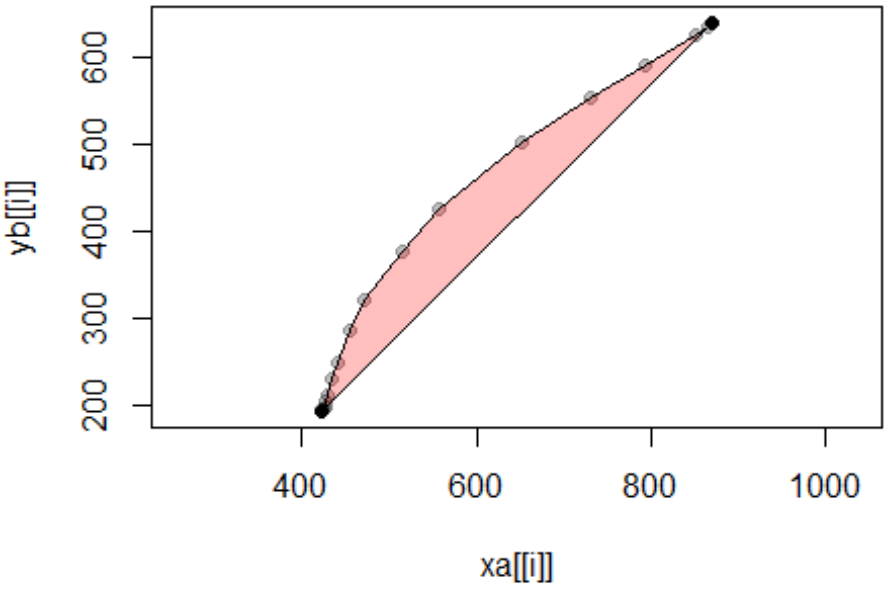

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

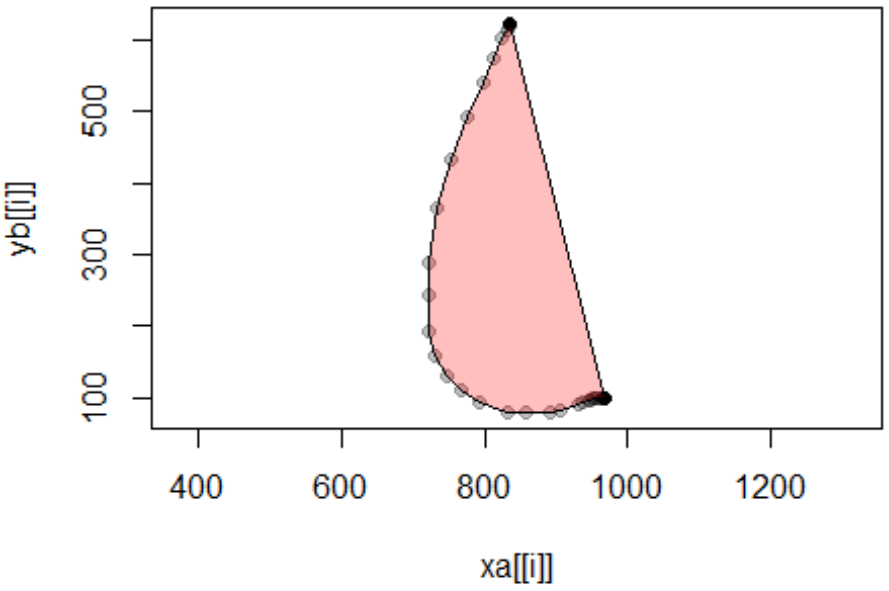

yb[[i]]
200
300
400
500
600
400
600
800
1000
1200
xa[[i]]

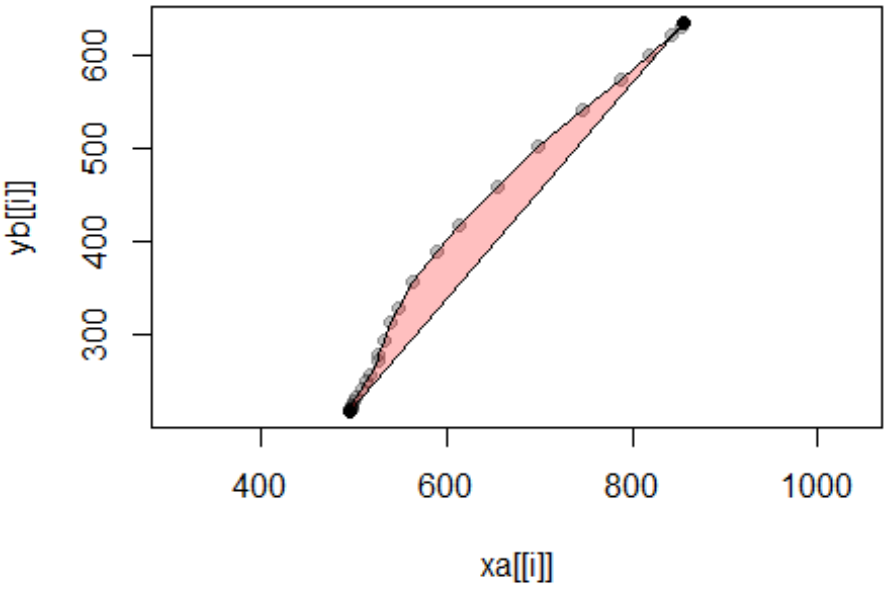

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

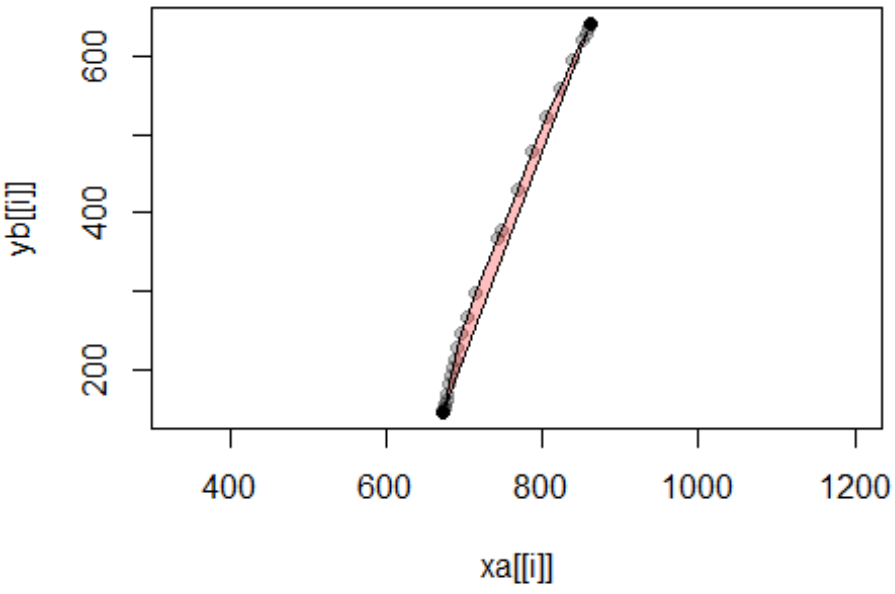

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

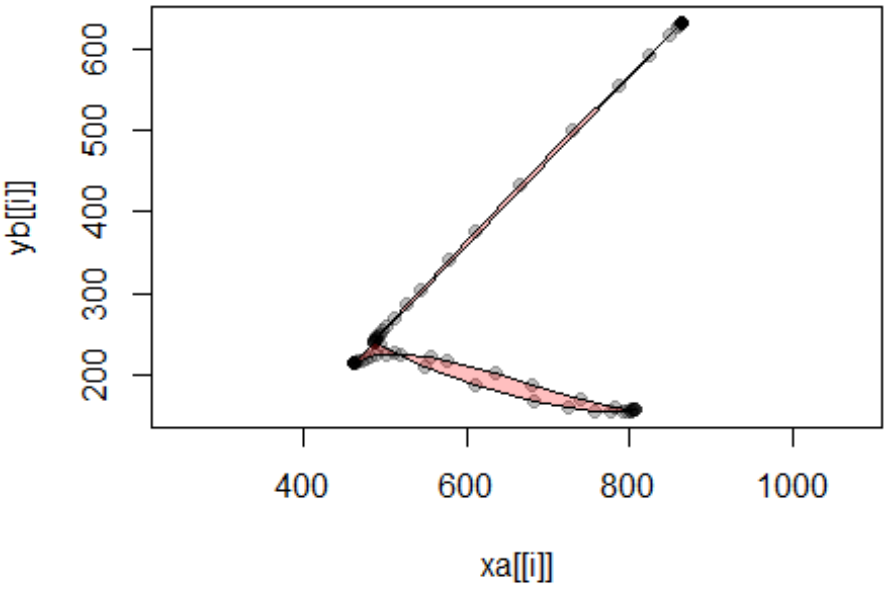

yb[[i]]
200
300
400
500
600
800
1000
1200
1400
xa[[i]]

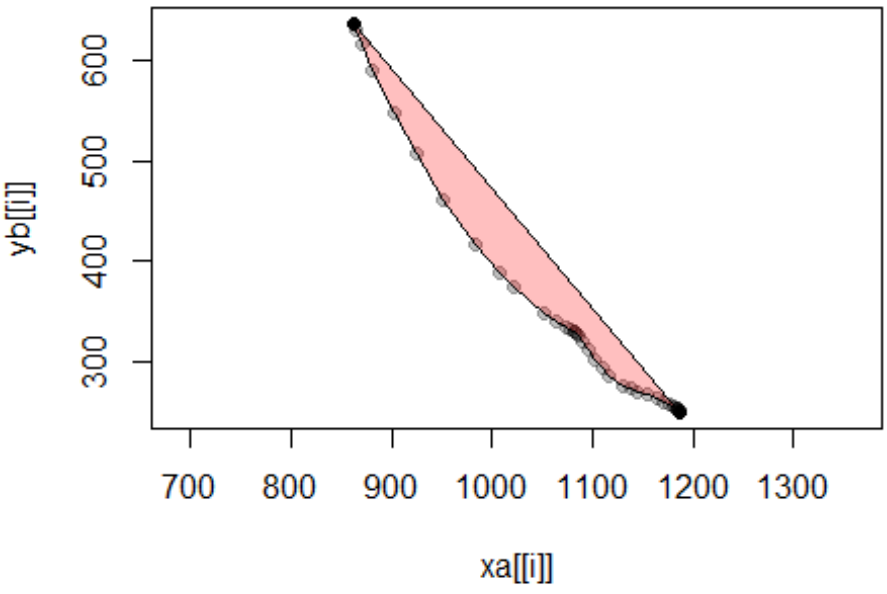

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

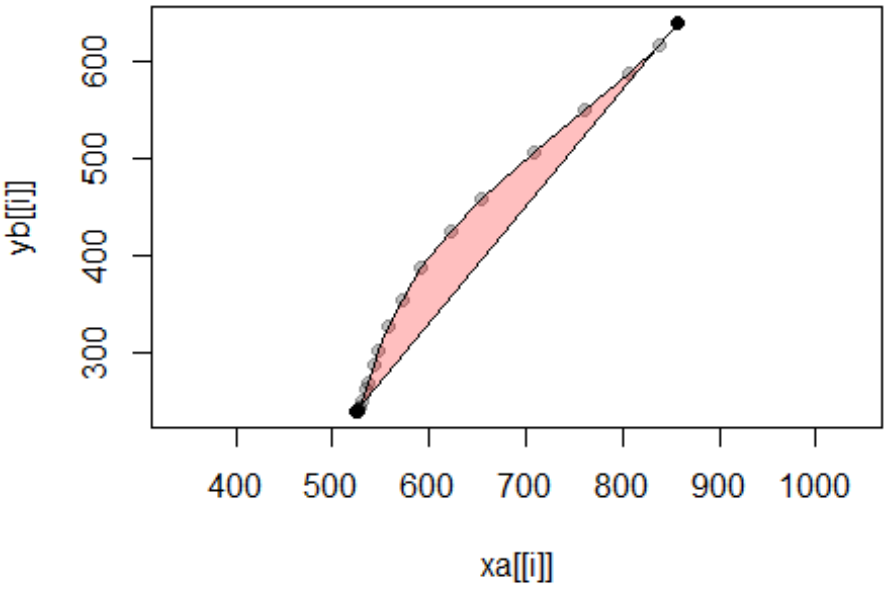

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

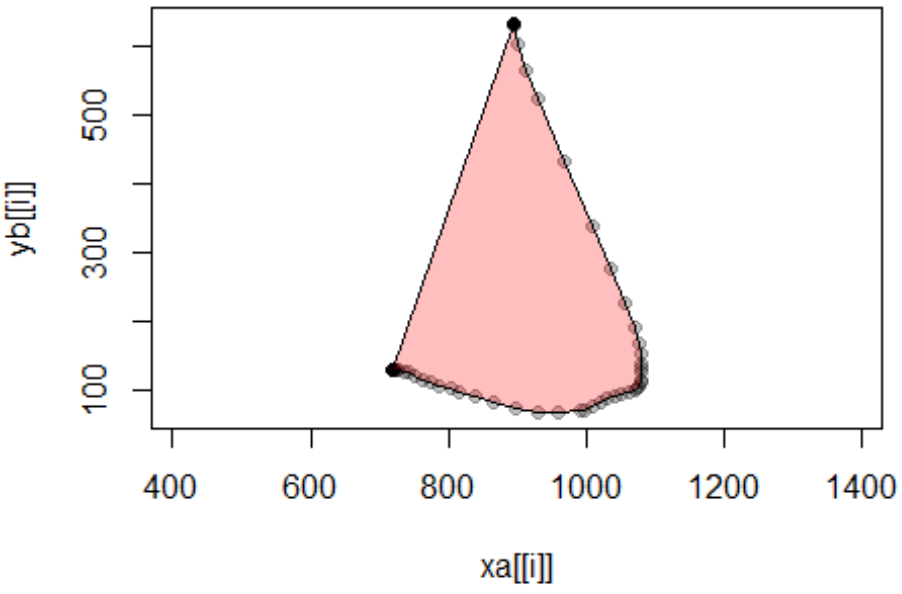

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

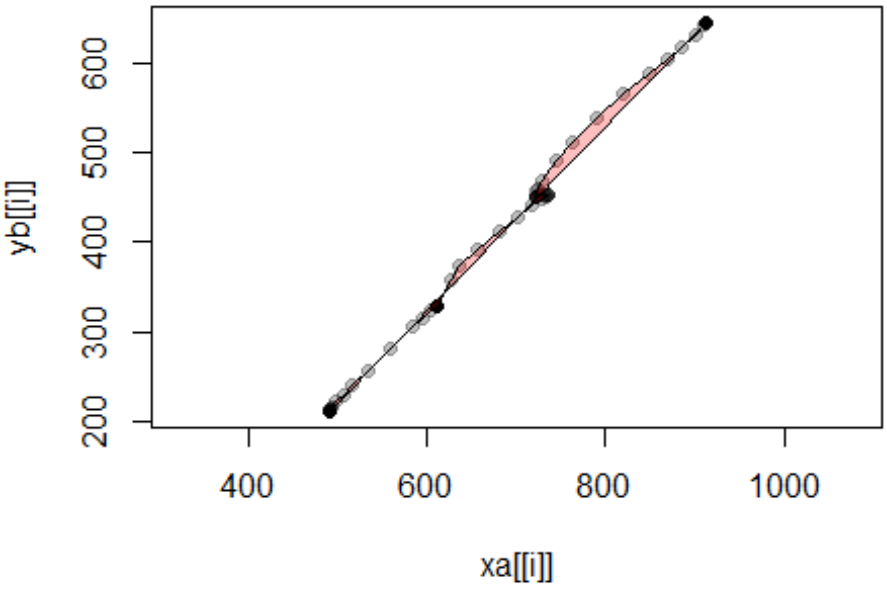

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

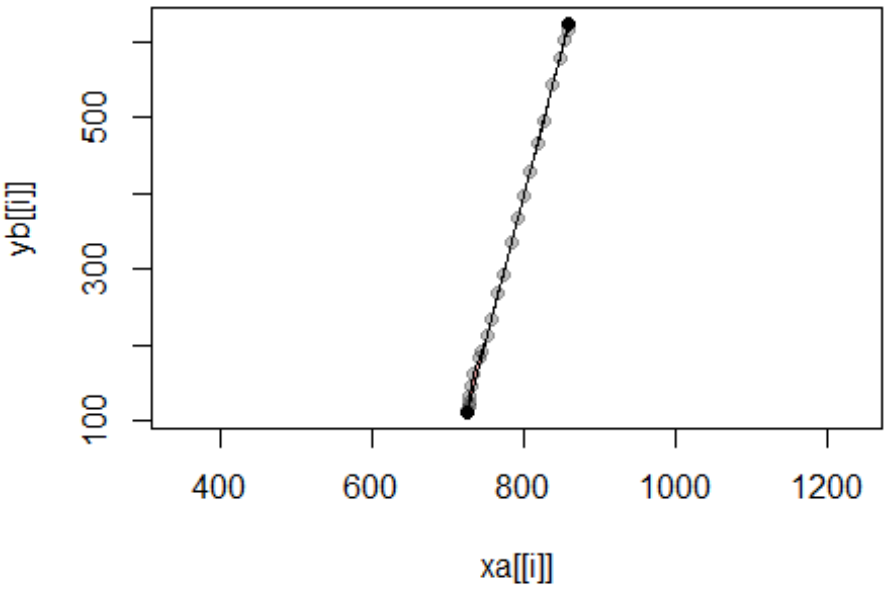

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

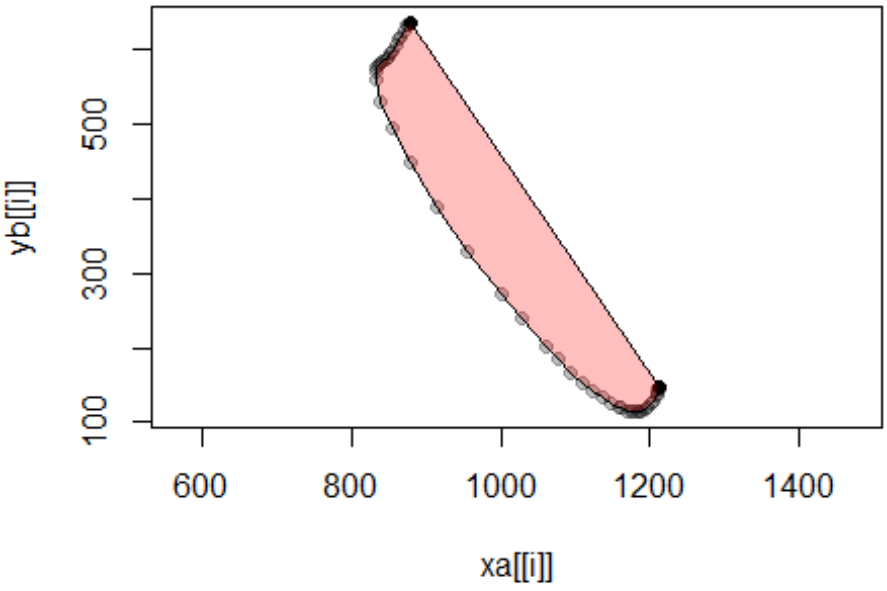

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
xa[[i]]

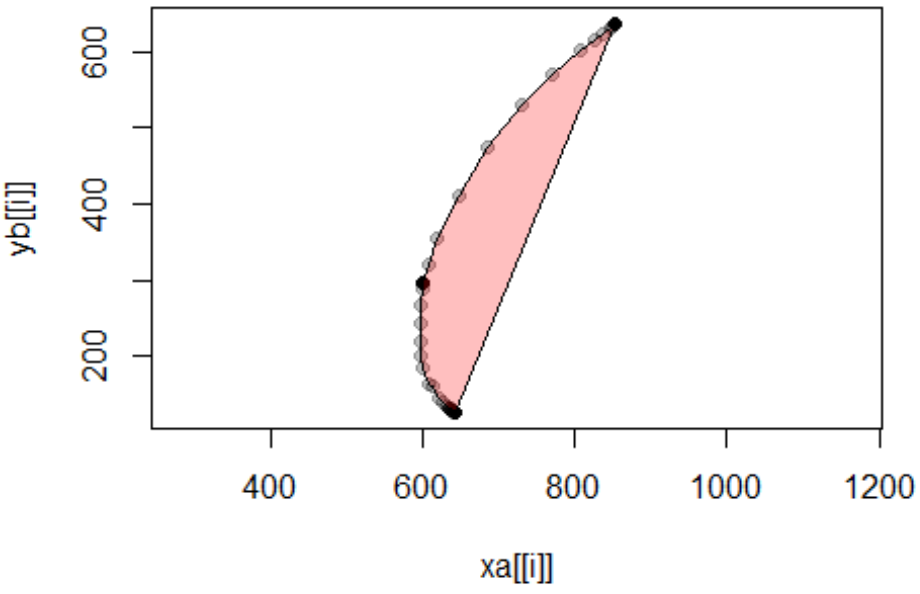

yb[[i]]
300
400
500
600
400
500
600
700
800
900
1000
xa[[i]]

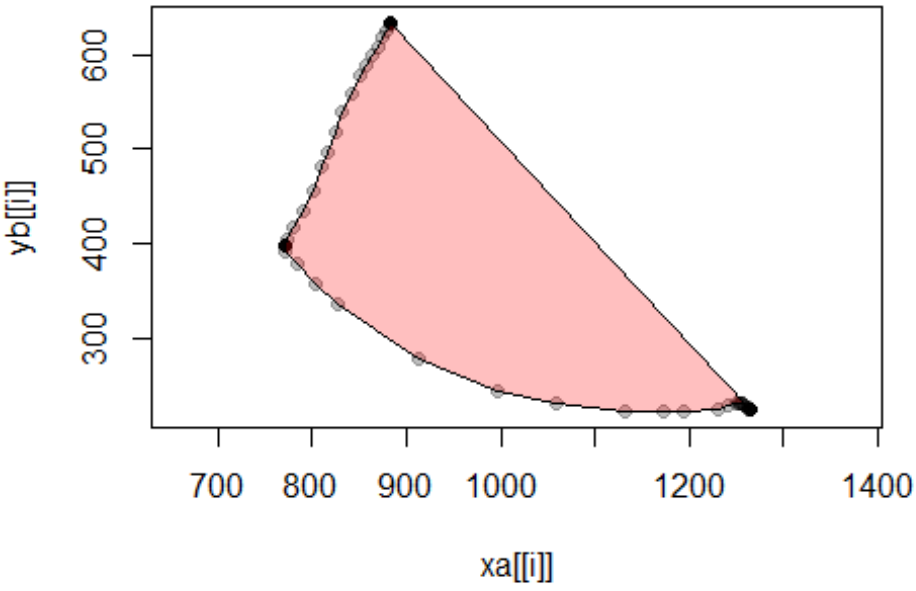

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

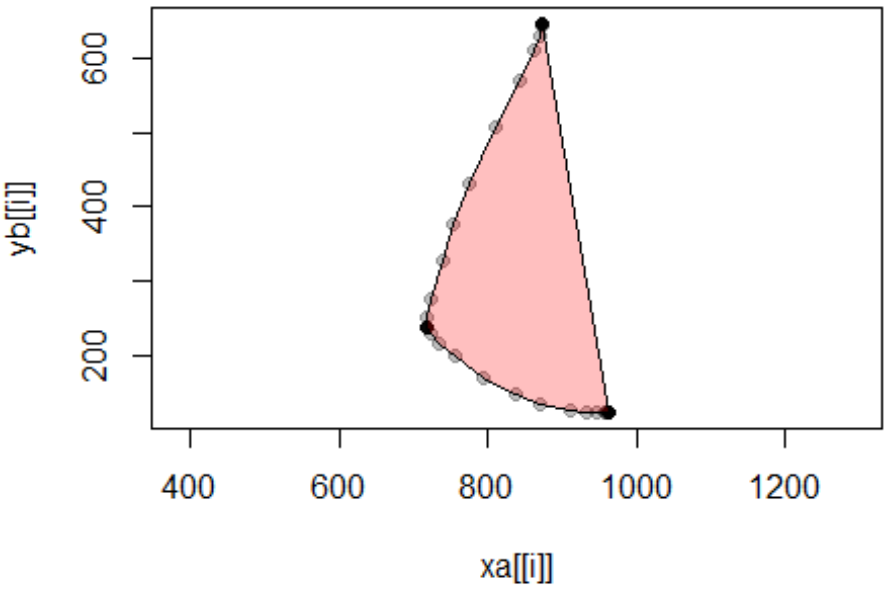

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

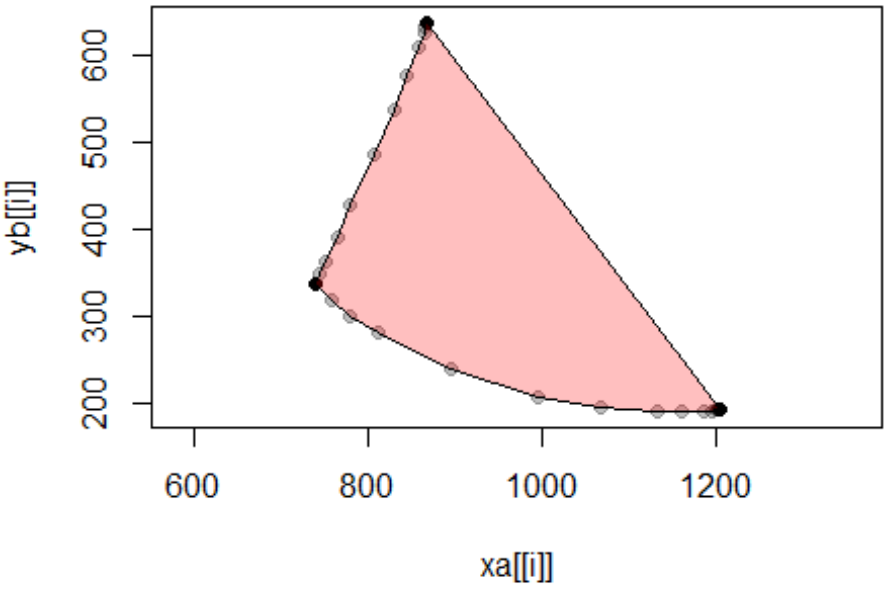

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

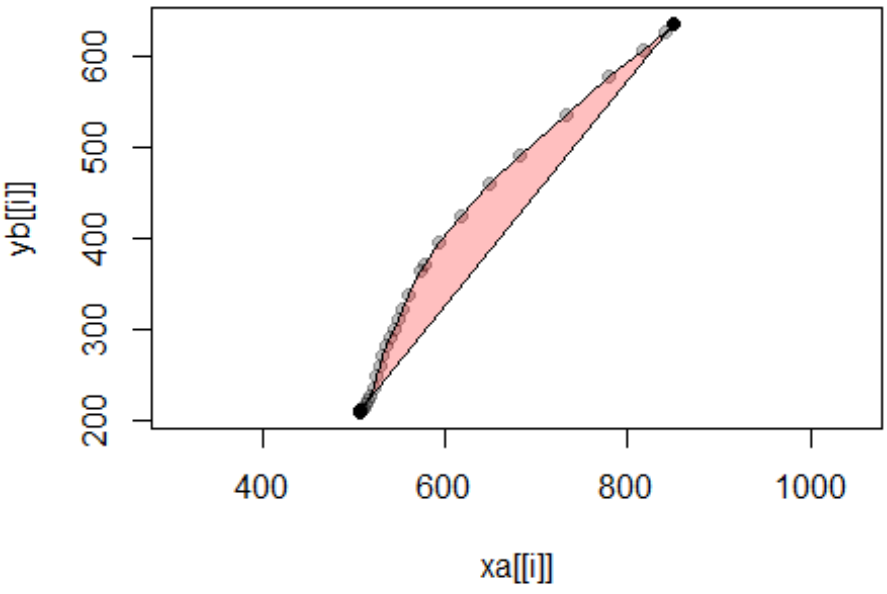

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

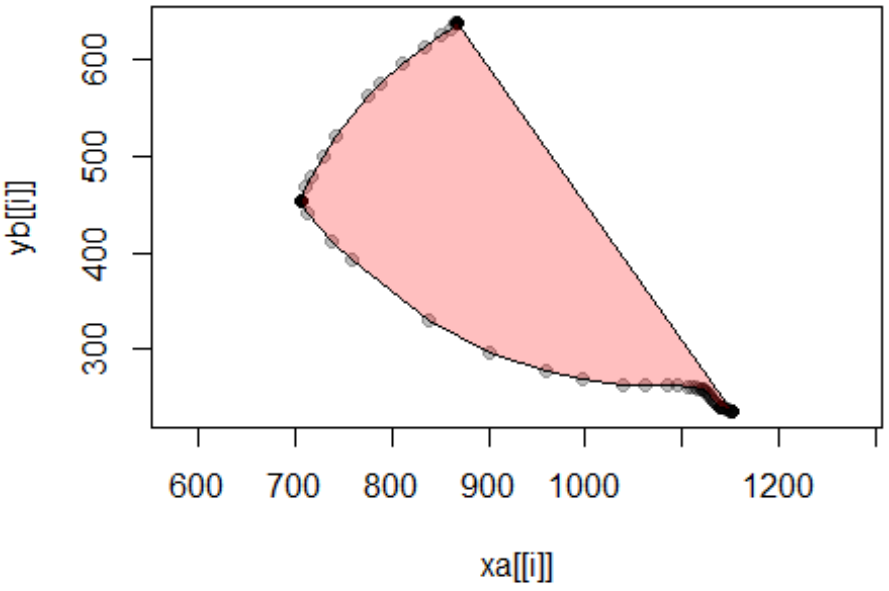

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

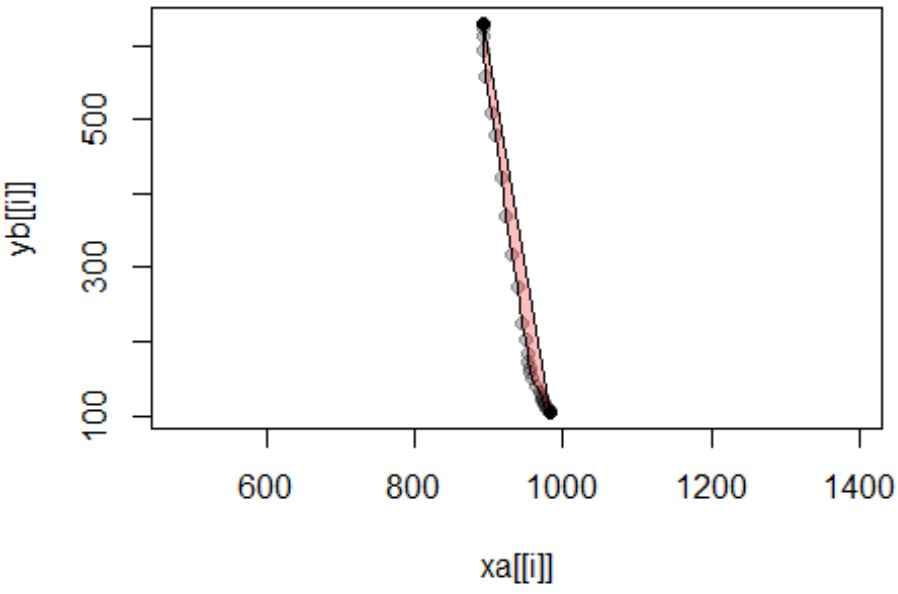

yb[[i]]
300
400
500
600
700
800
900
1000
1200
1400
xa[[i]]

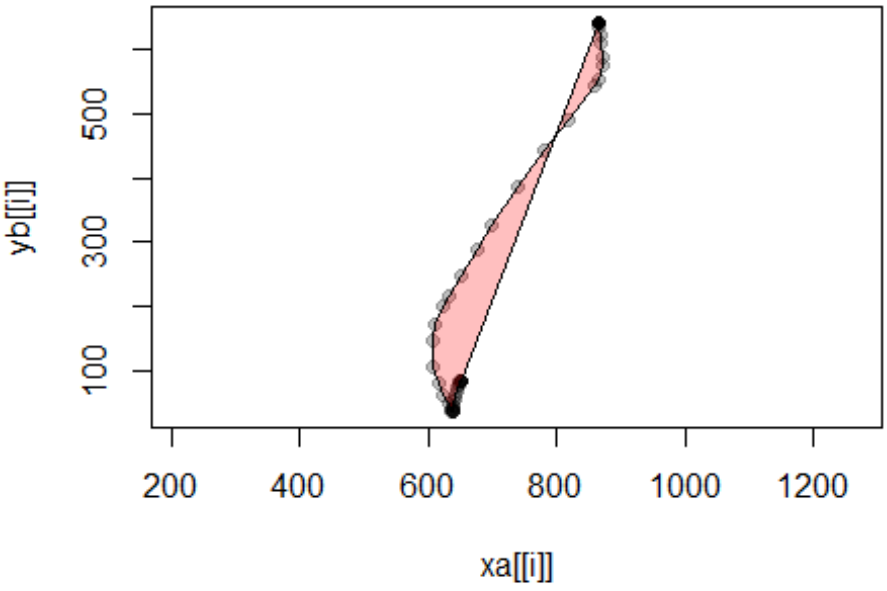

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

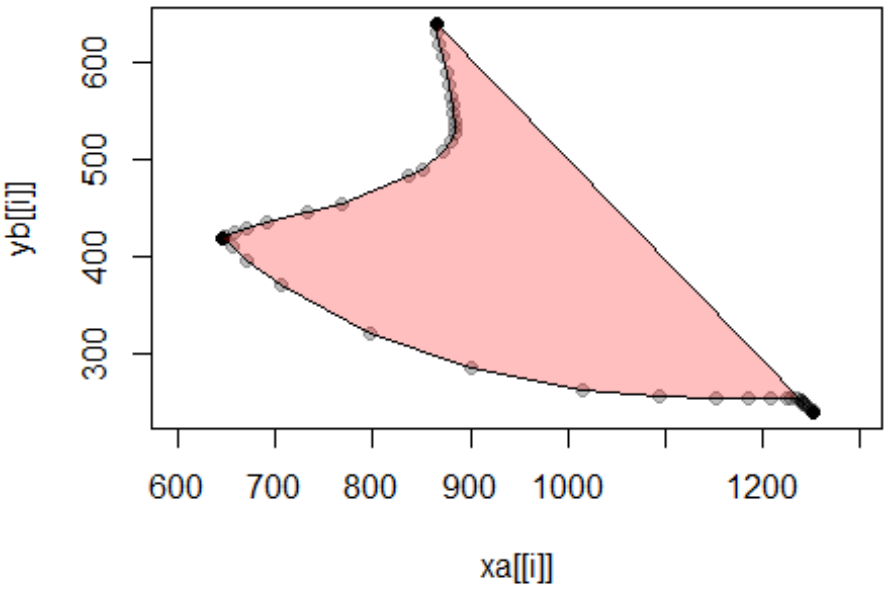

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

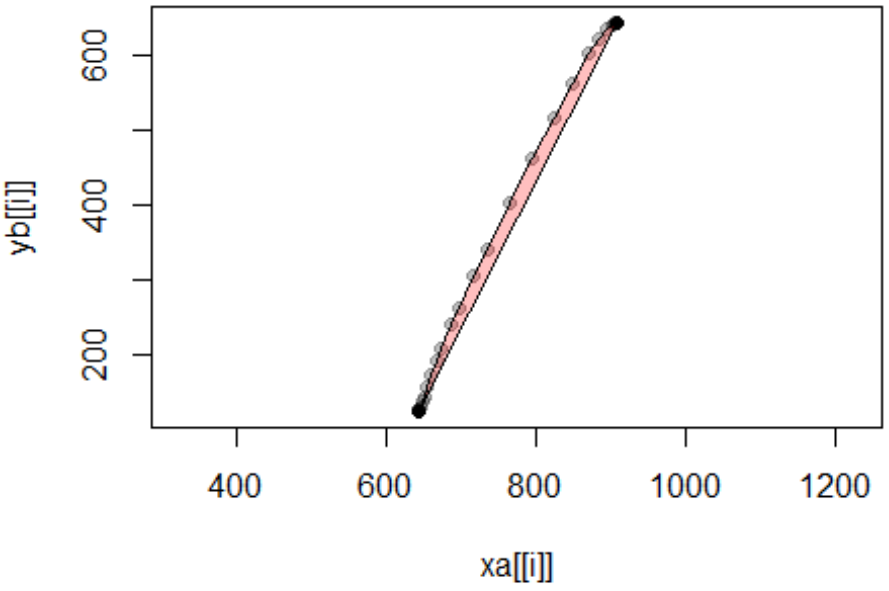

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

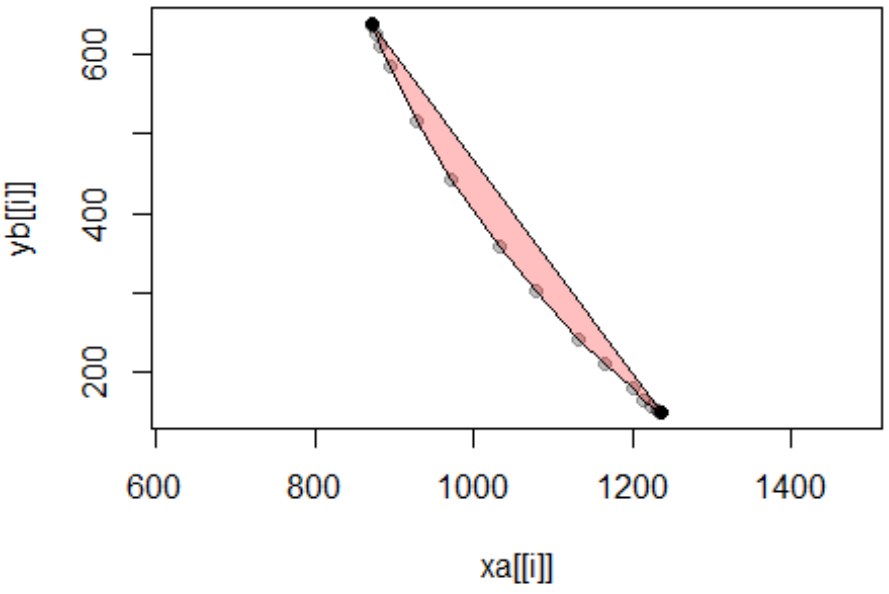

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

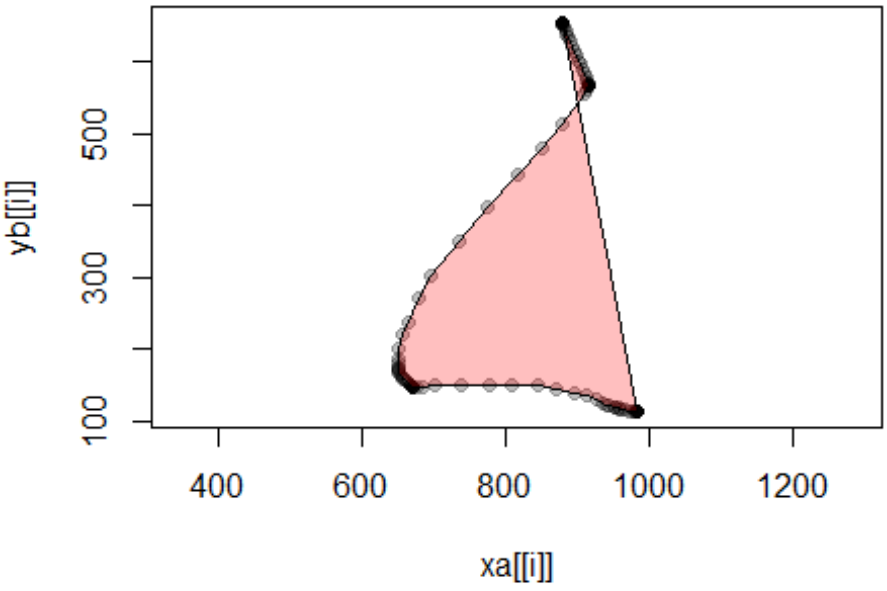

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

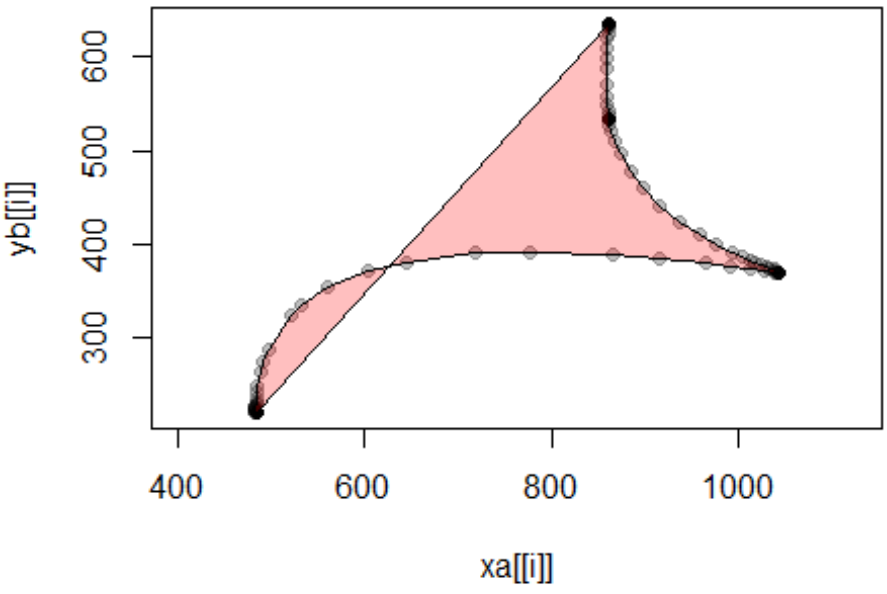

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

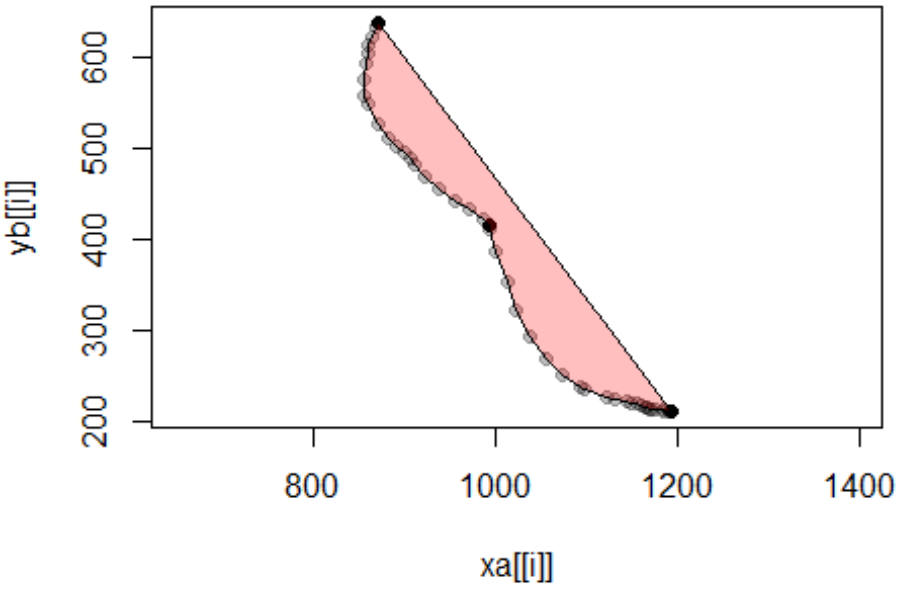

yb[[i]]
300
400
500
600
600
700
800
900
1000
1200
xa[[i]]

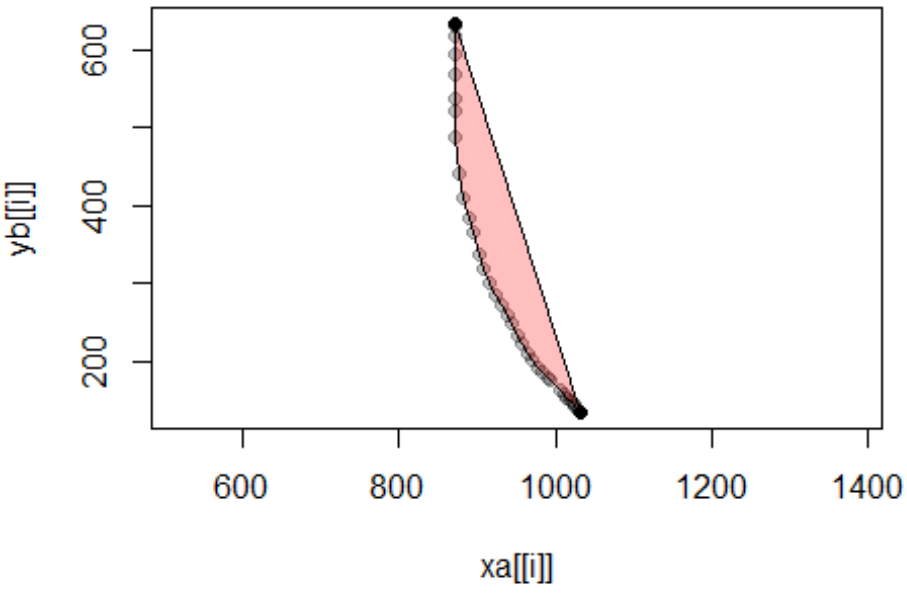

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

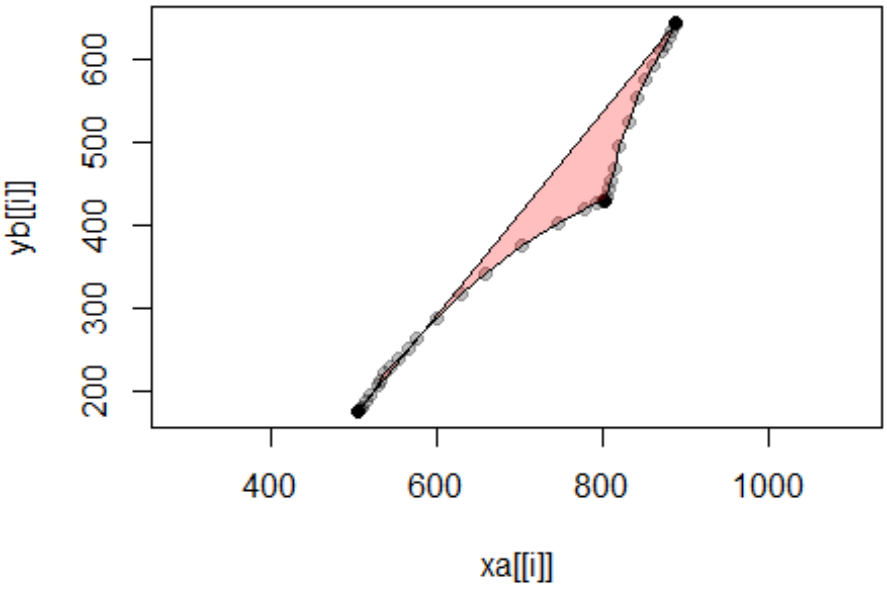

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

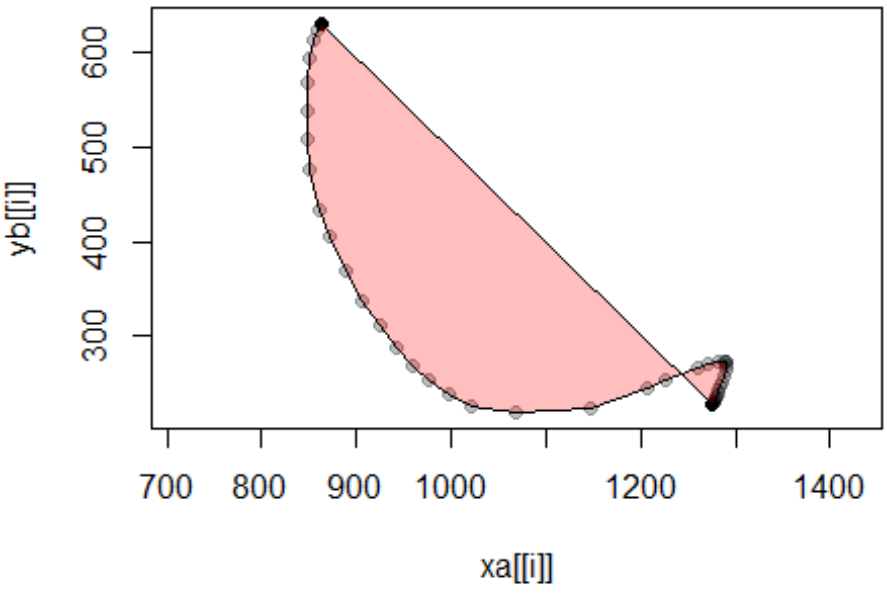

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

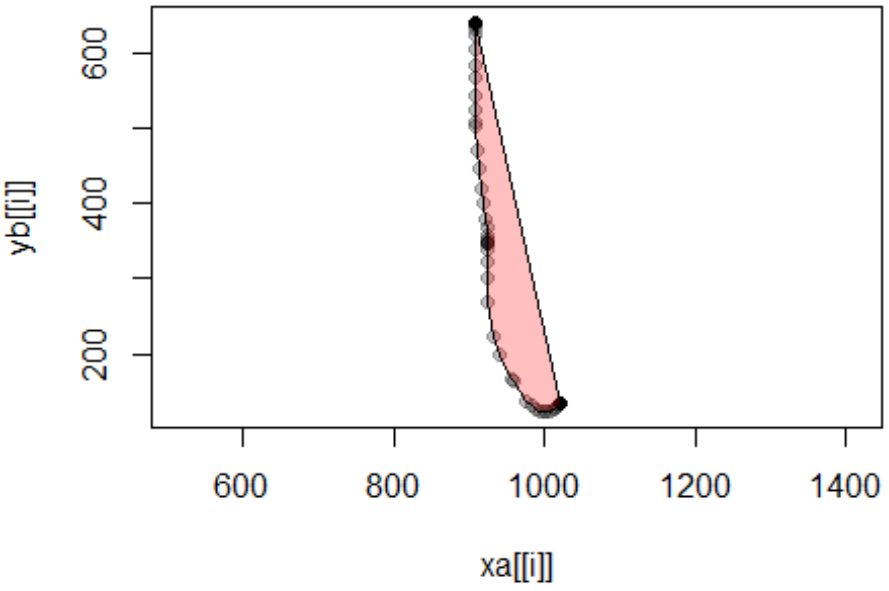

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

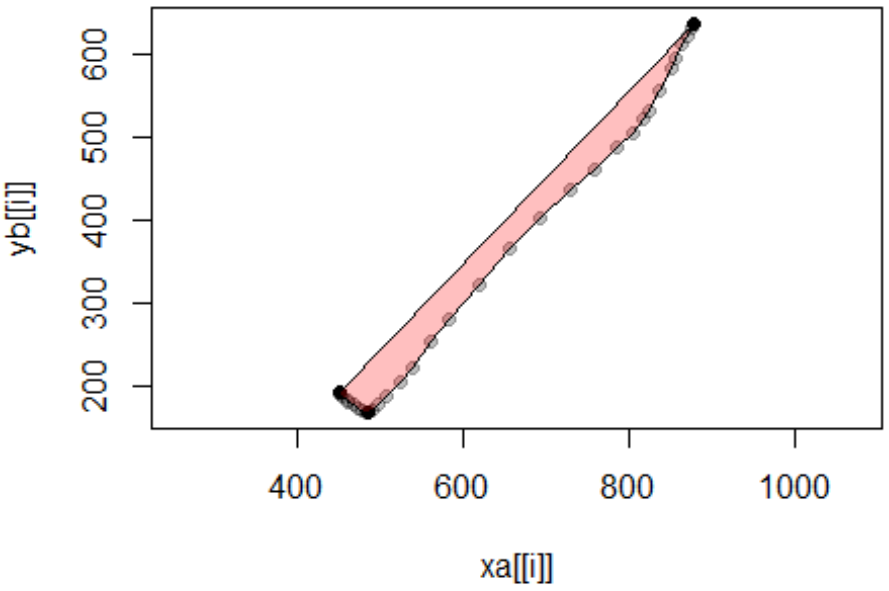

yb[[i]]
200
300
400
500
600
800
1000
1200
1400
xa[[i]]

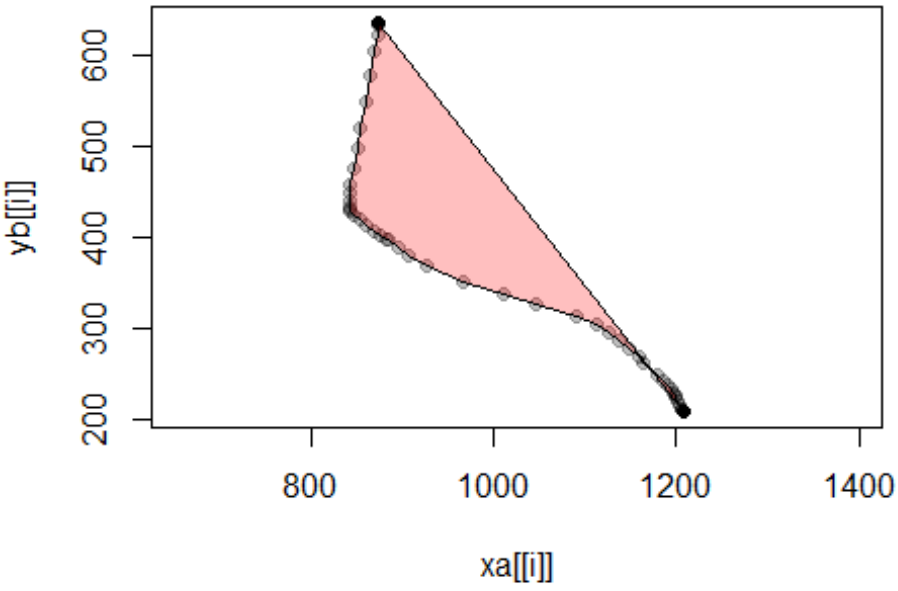

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

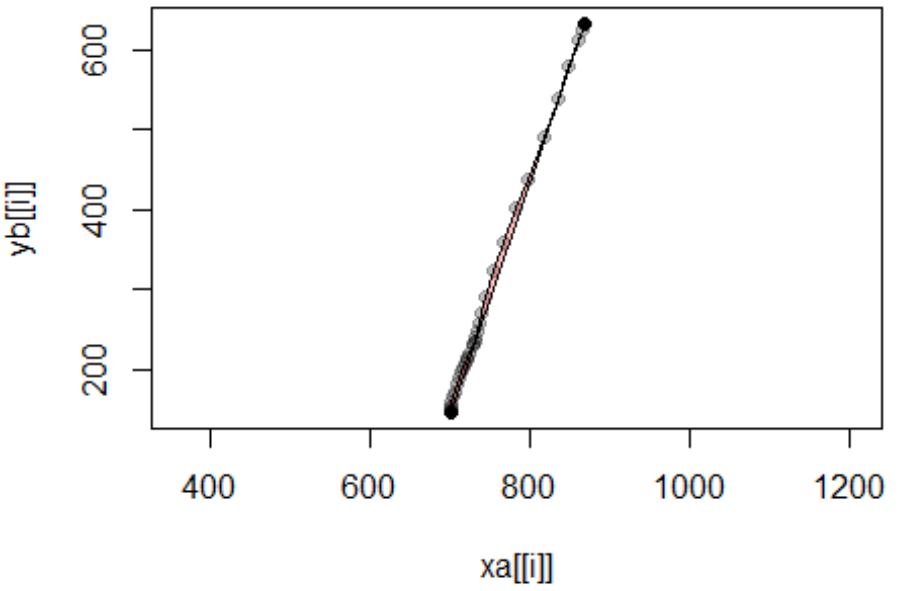

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

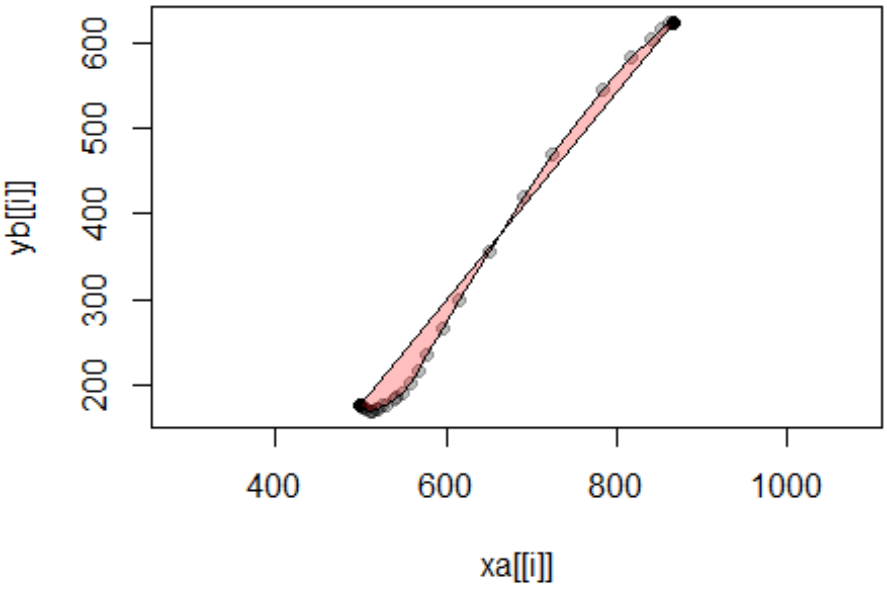

yb[[i]]
300
400
500
600
700
800
900
1000
1200
1400
xa[[i]]

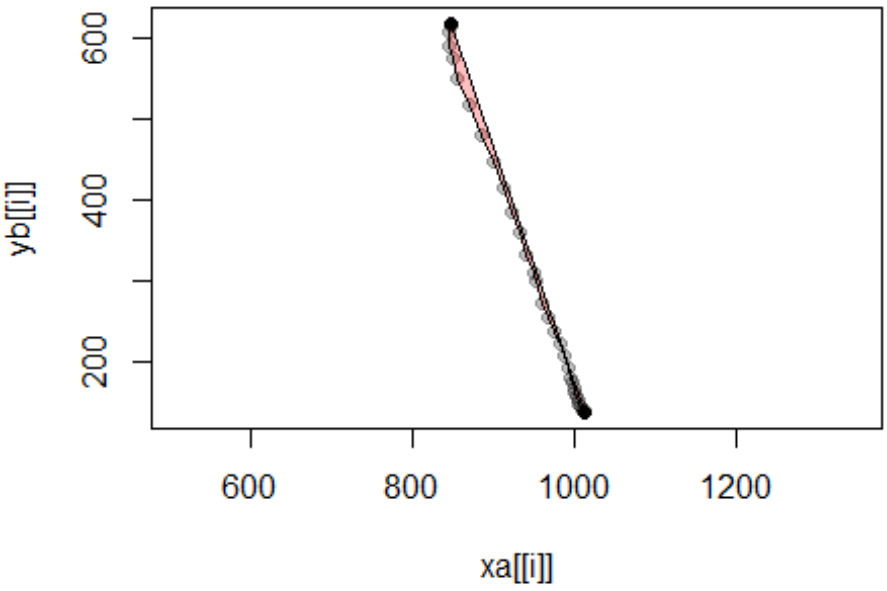

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

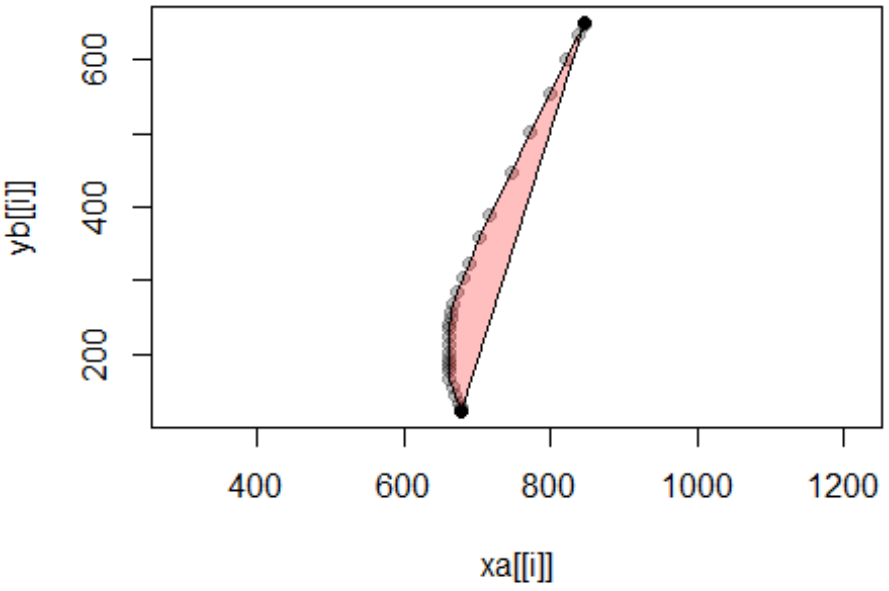

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

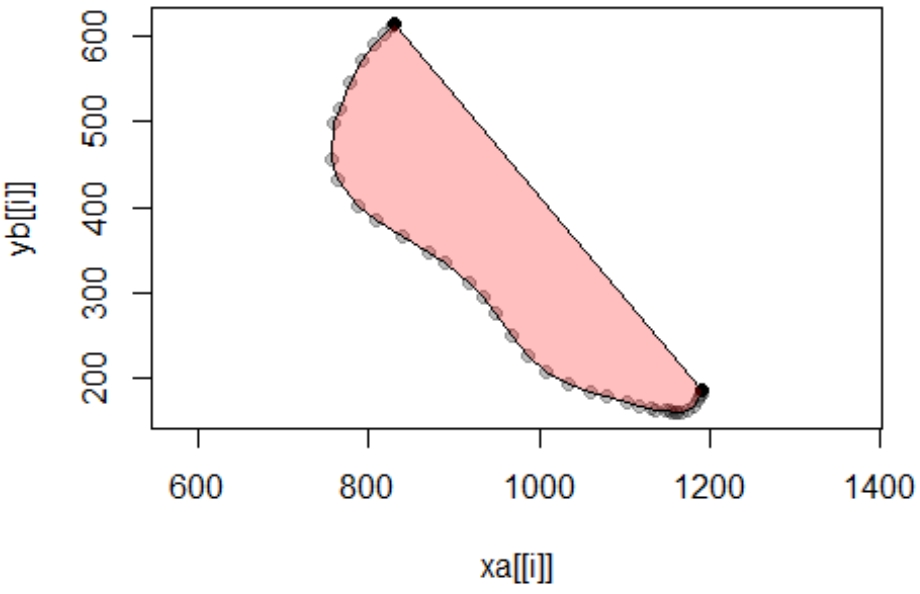

yb[[i]]
200
300
400
500
600
800
1000
1200
1400
xa[[i]]

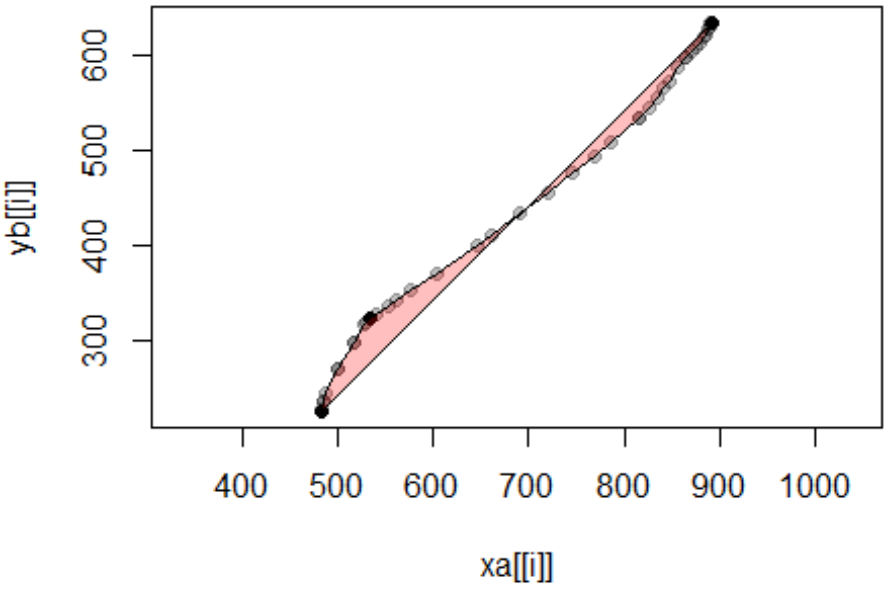

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

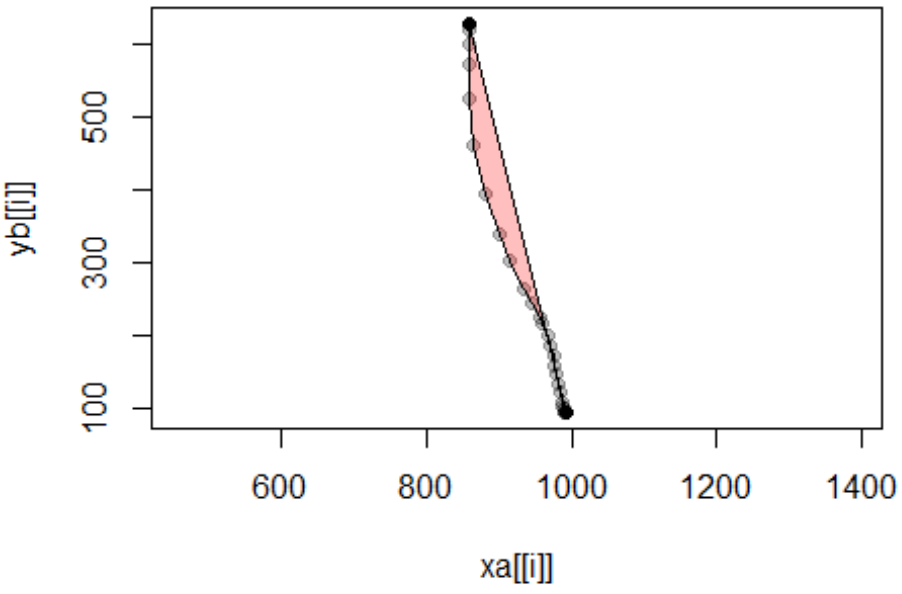

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

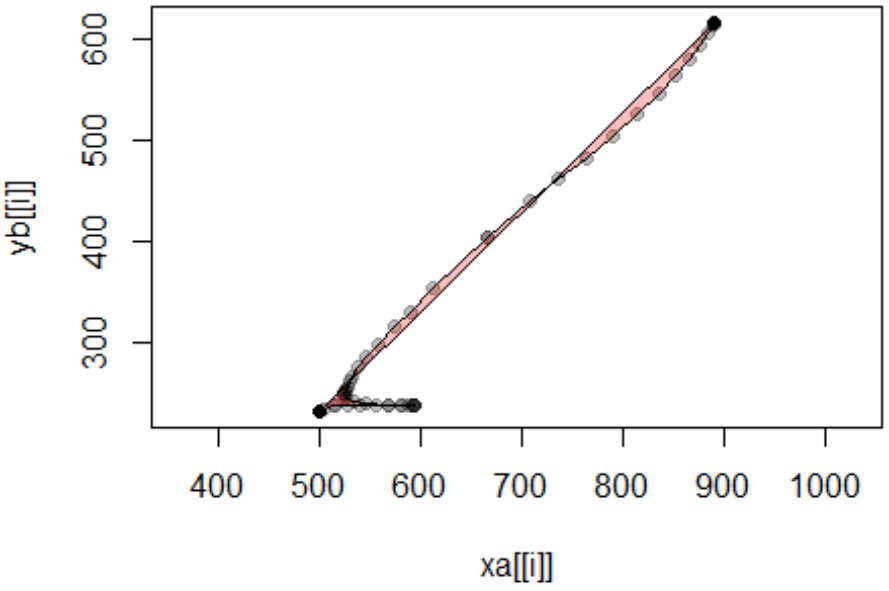

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

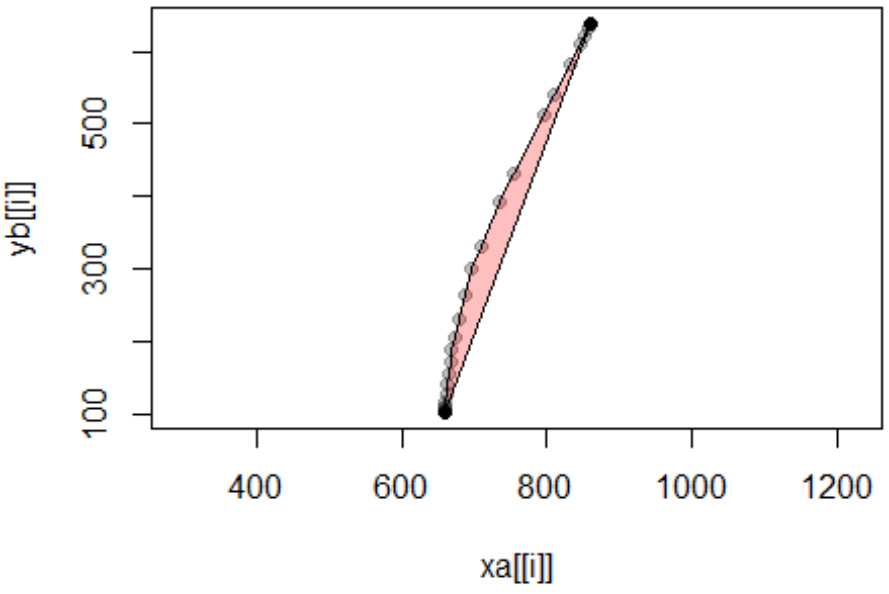

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

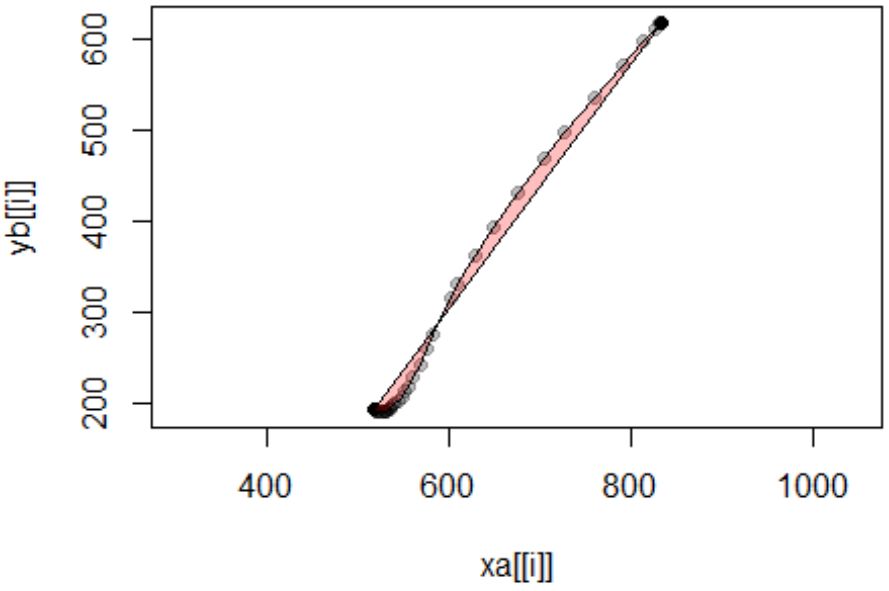

yb[[i]]
200
300
400
500
600
600
800
1000
1200
1400
xa[[i]]

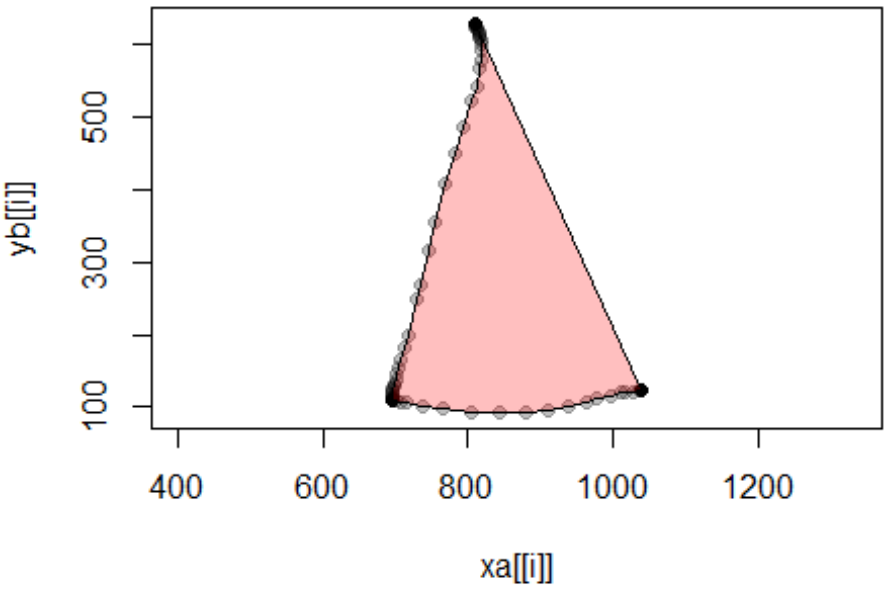

yb[[i]]
300
400
500
600
400
500
600
700
800
900
1000
xa[[i]]

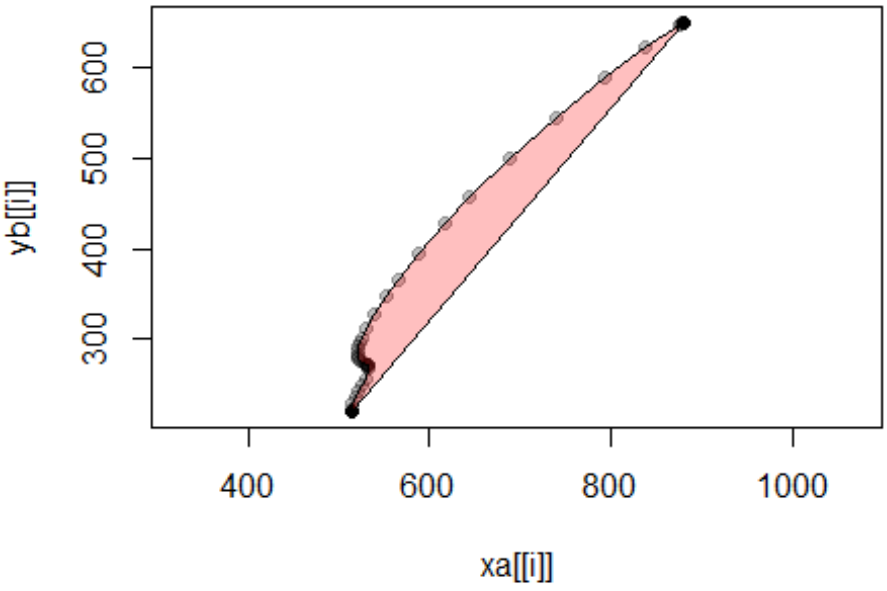

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

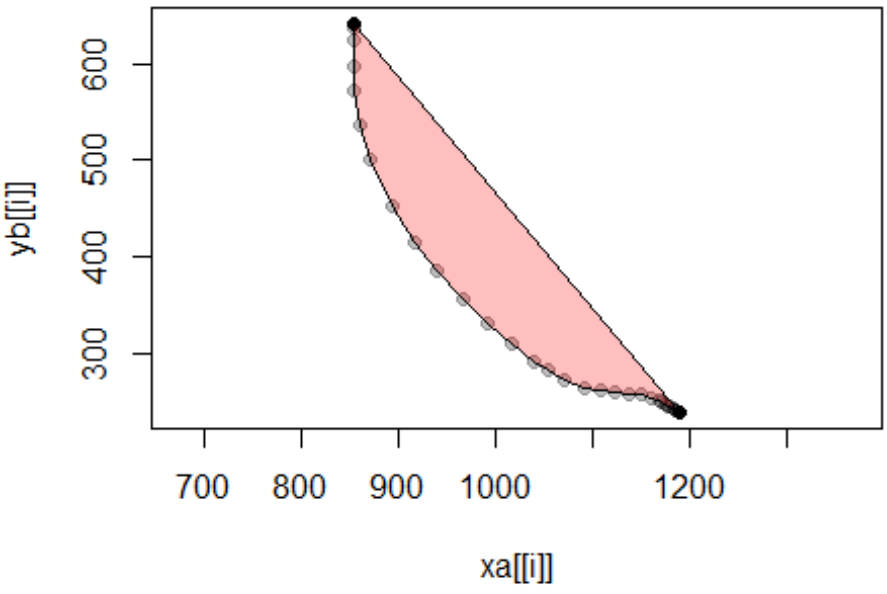

yb[[i]]
300
400
500
600
400
500
600
700
800
900
1000
xa[[i]]

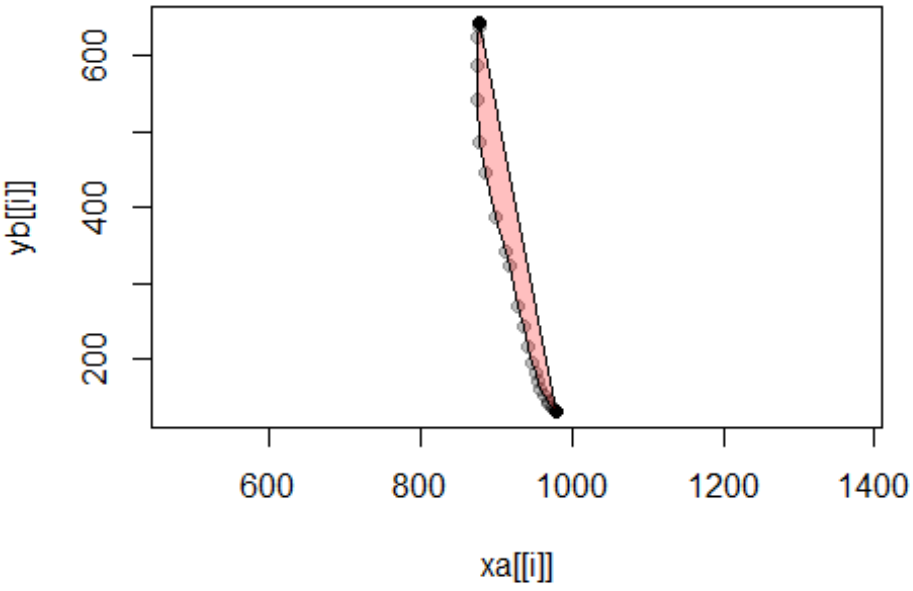

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

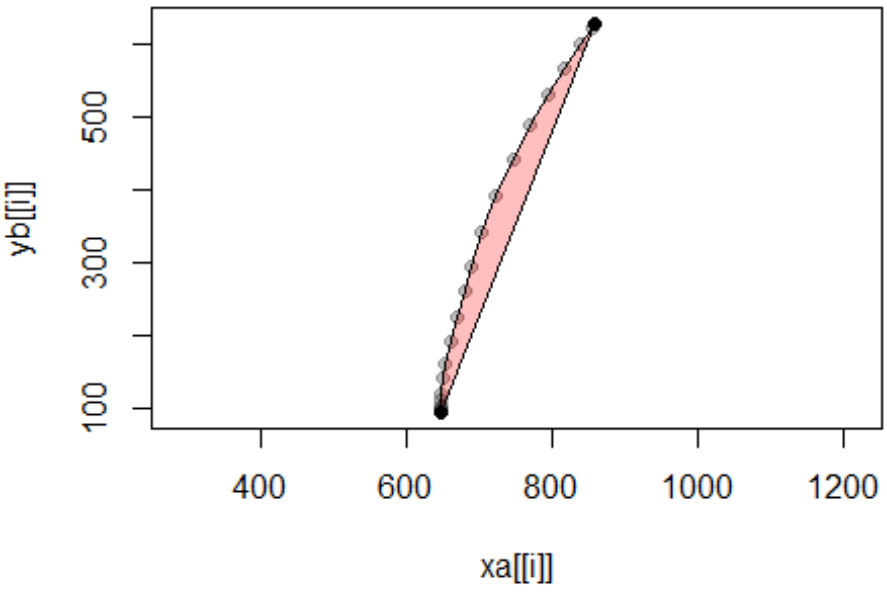

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

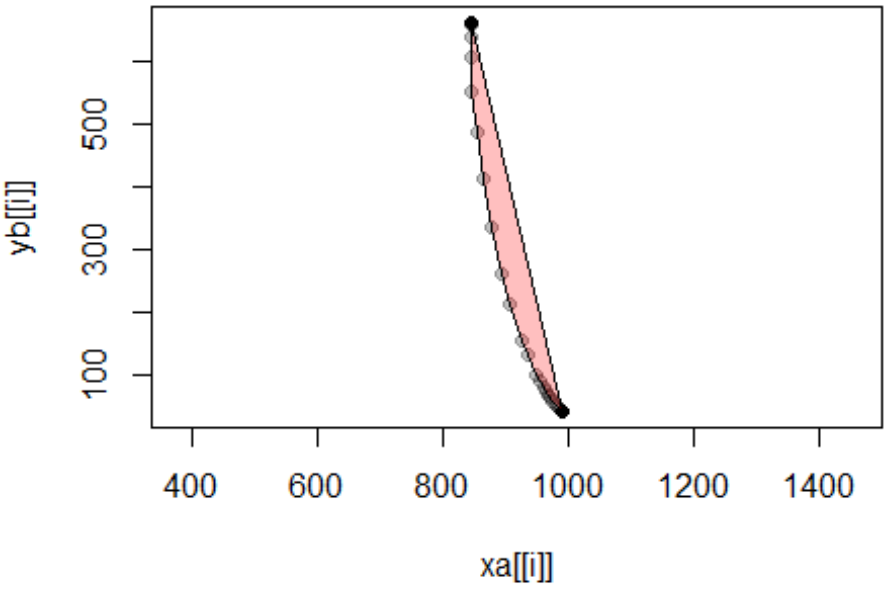

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

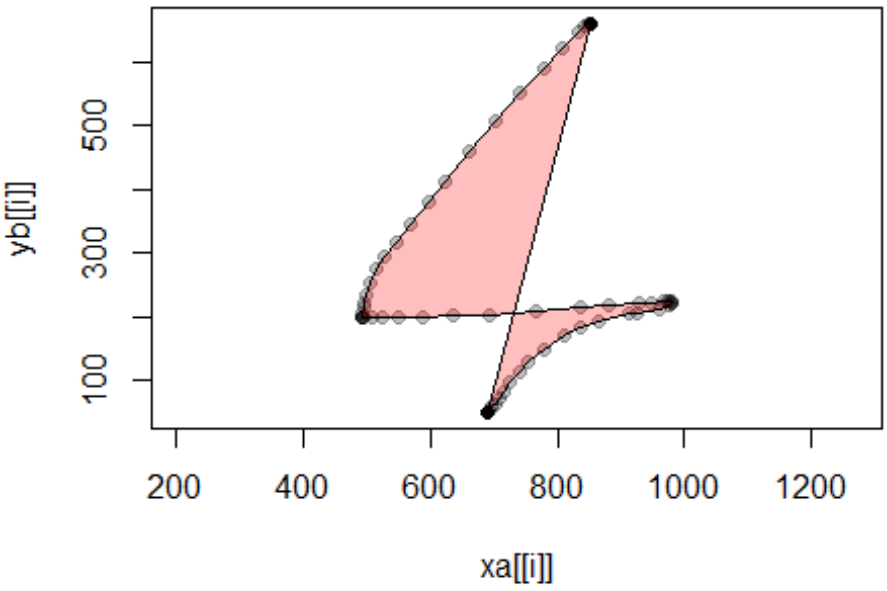

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

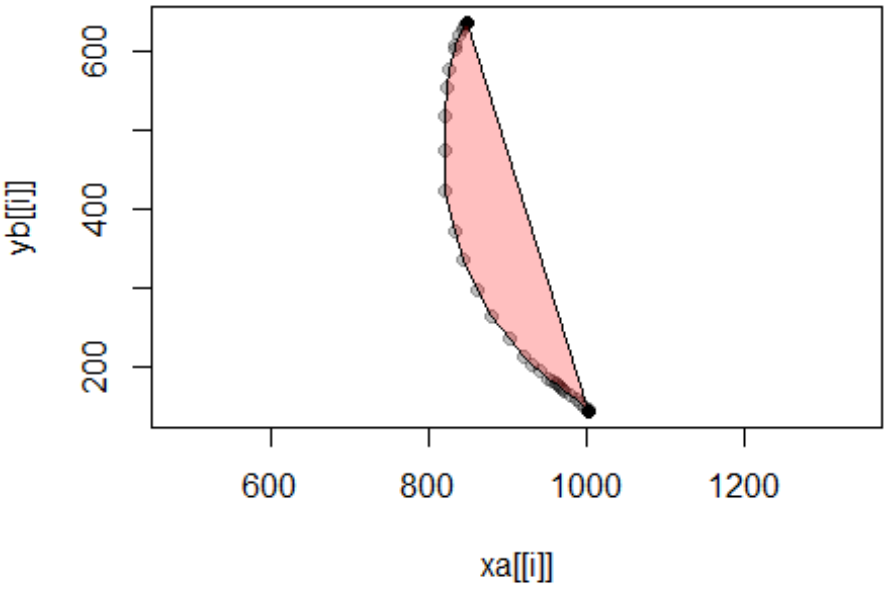

yb[[i]]
300
400
500
600
700
800
900
1000
1200
xa[[i]]

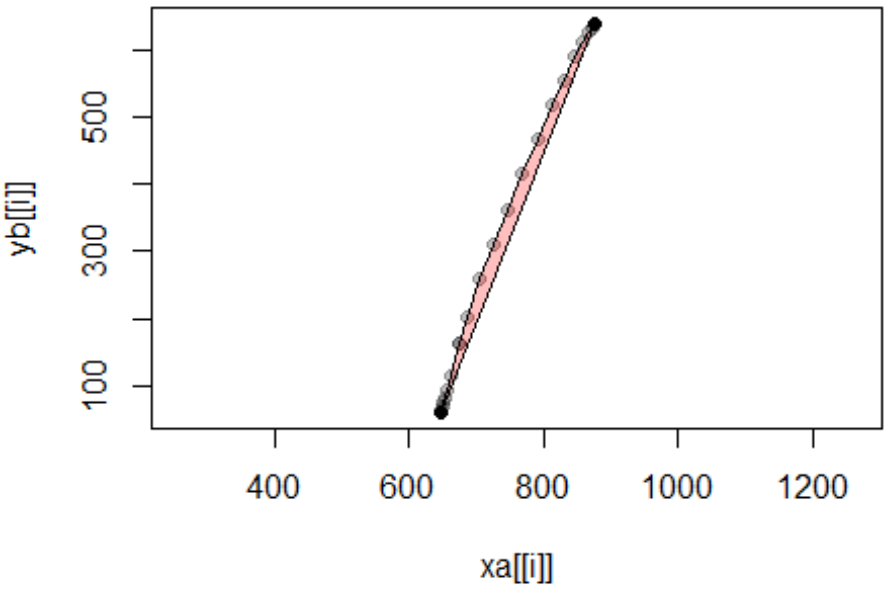

yb[[i]]
600
400
200
600
800
1000
1200
1400
xa[[i]]

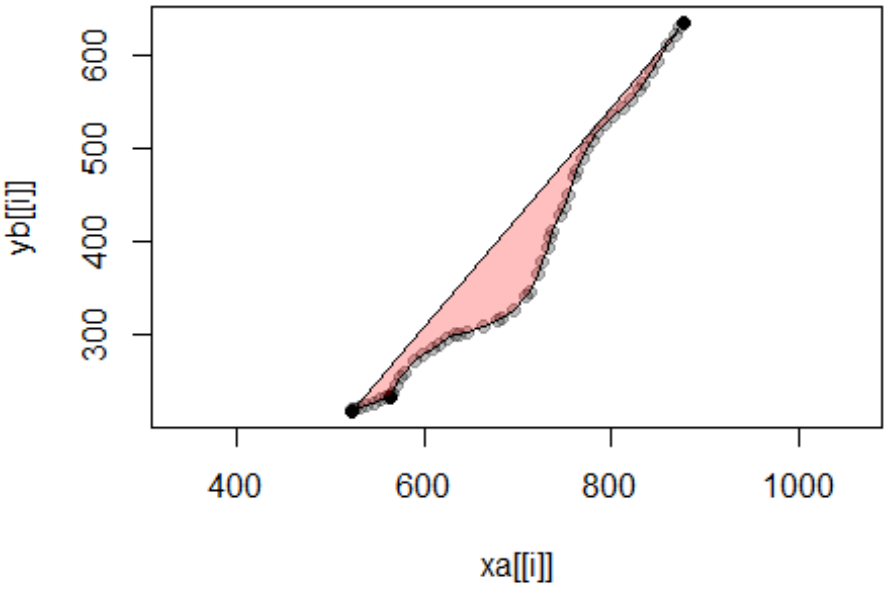

yb[[i]]
500
300
100
400
600
800
1000
1200
xa[[i]]

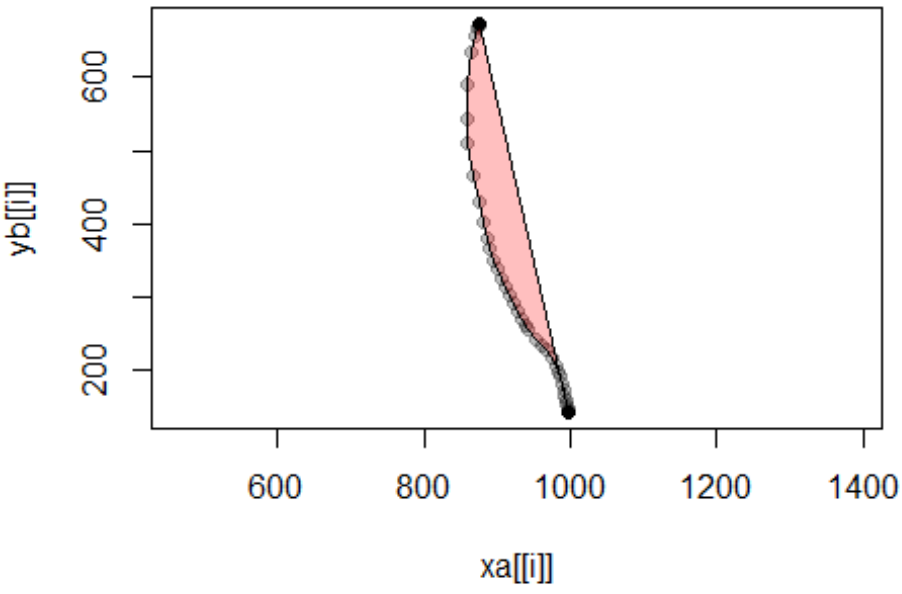

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

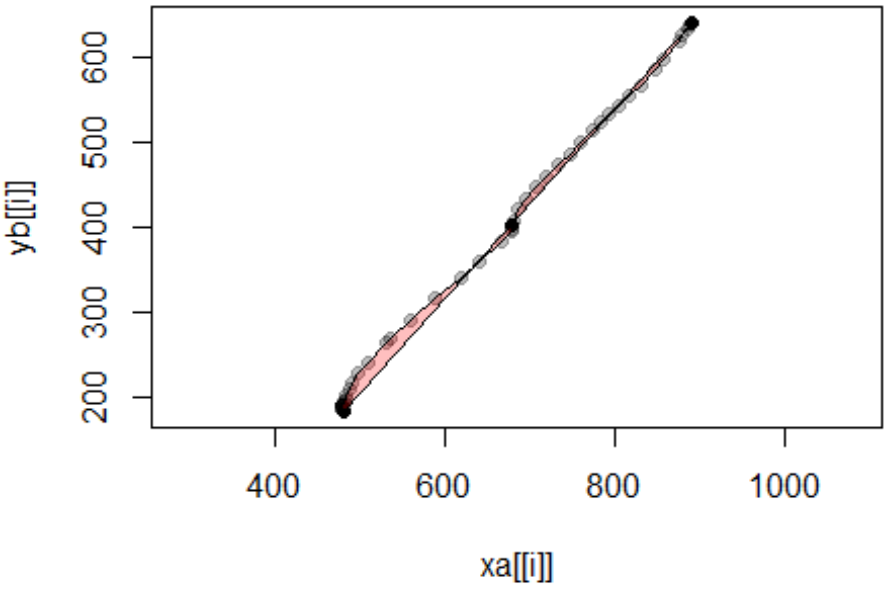

yb[[i]]
100
300
500
200
400
600
800
1000
1200
xa[[i]]

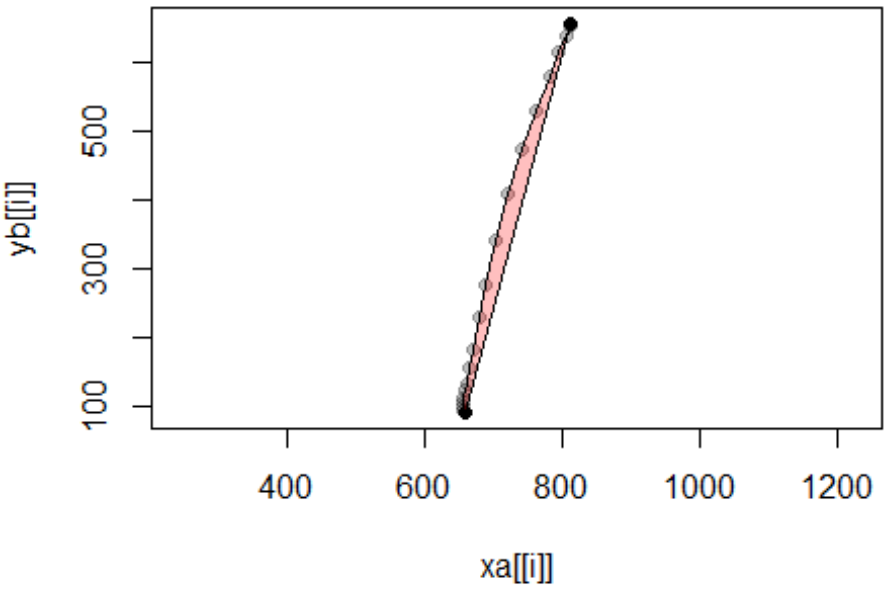

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

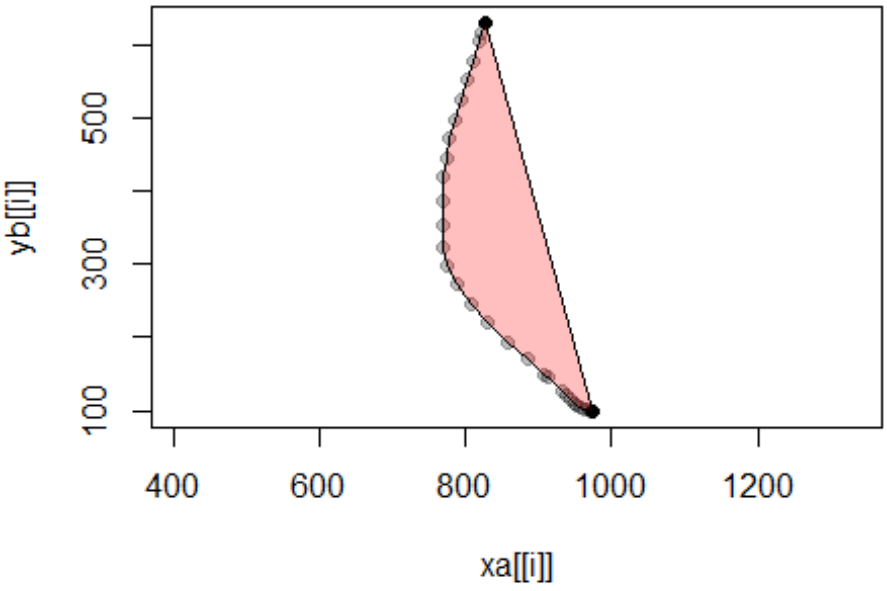

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

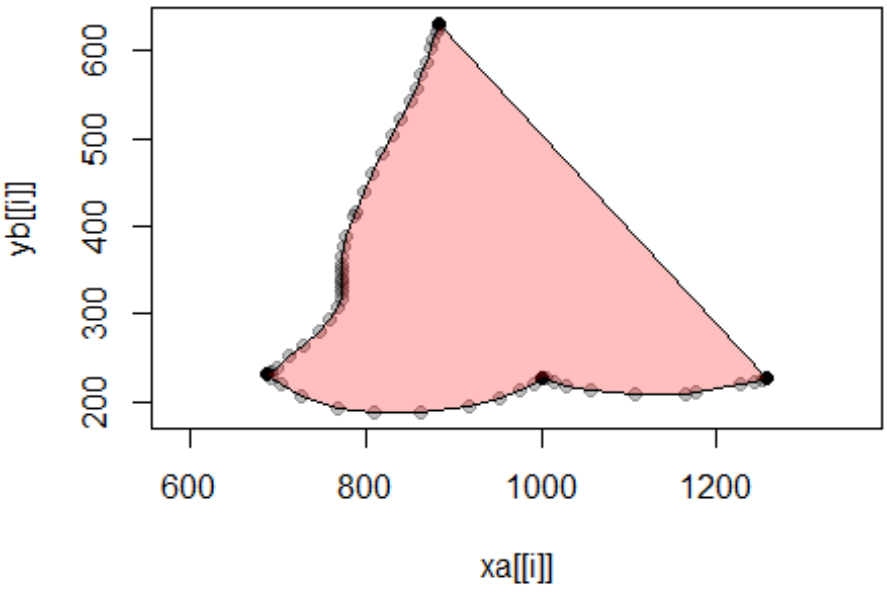

yb[[i]]
300
400
500
600
400
600
800
1000
xa[[i]]

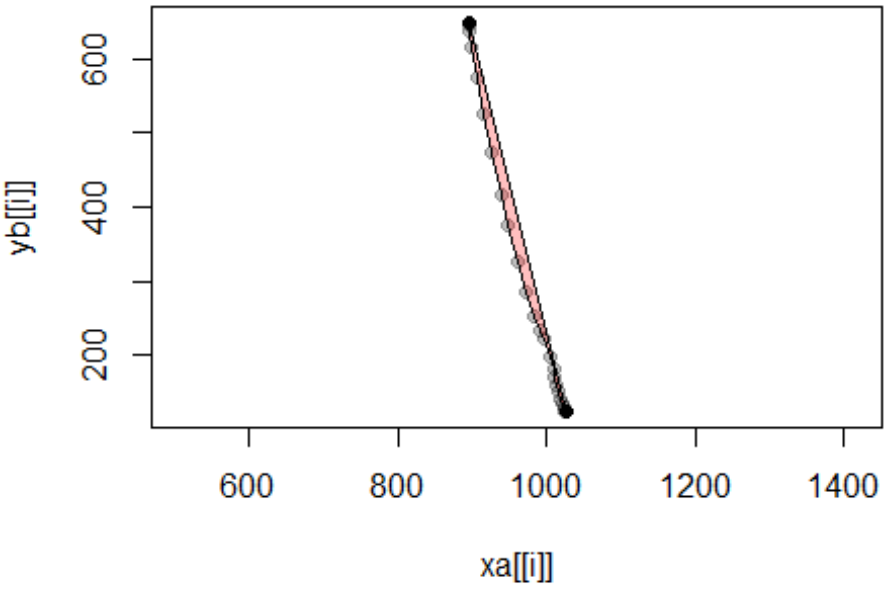

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

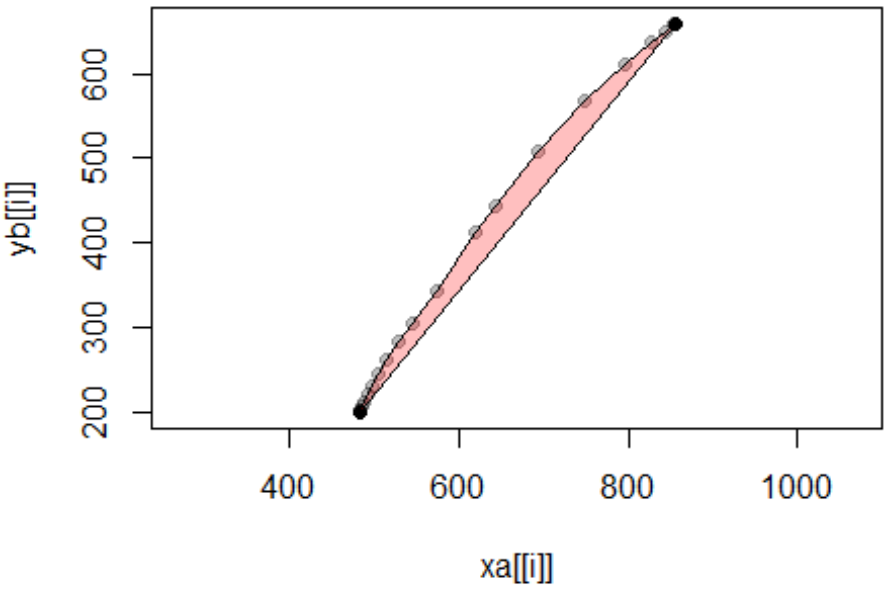

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

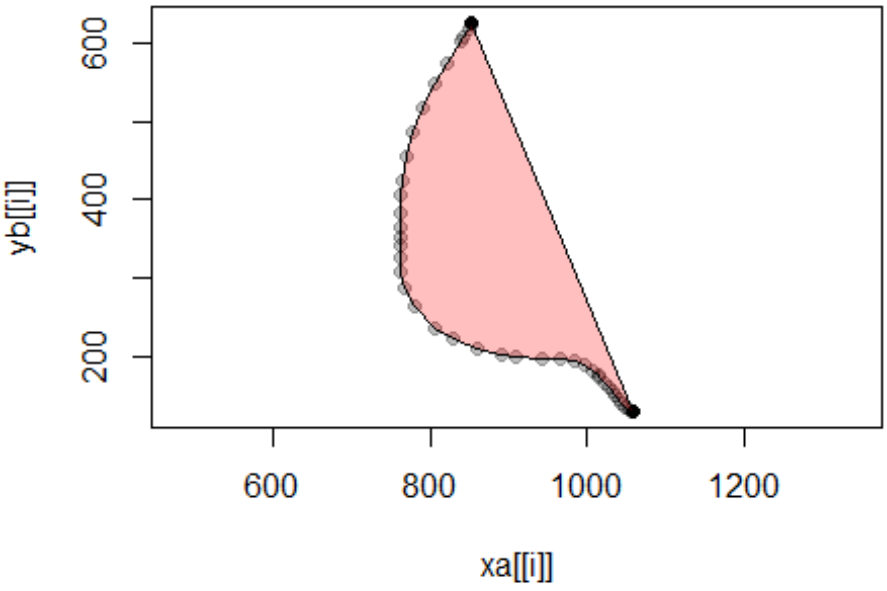

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

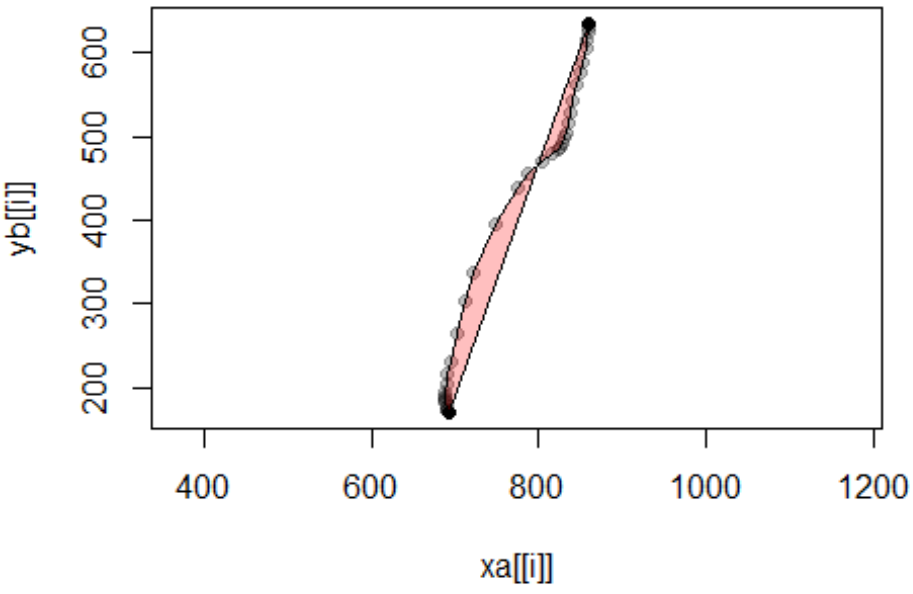

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

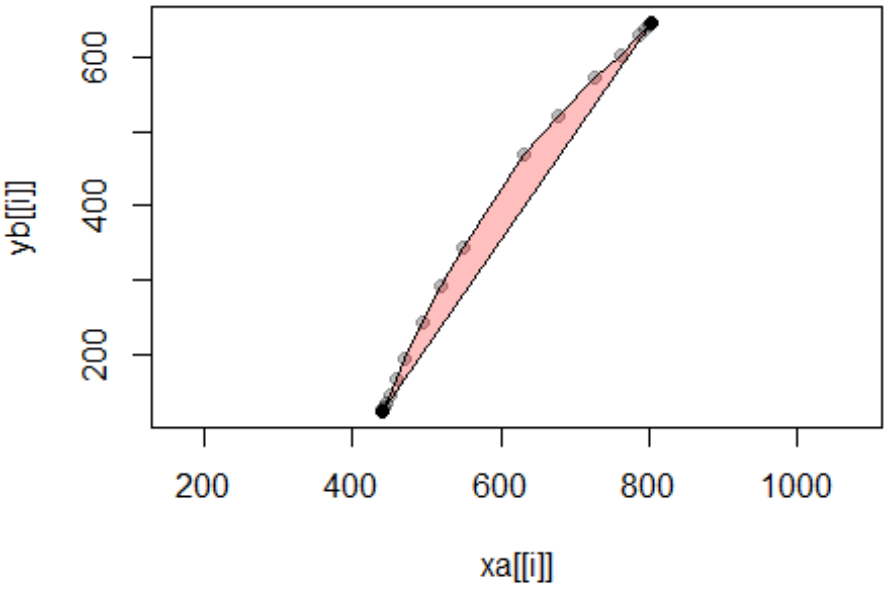

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

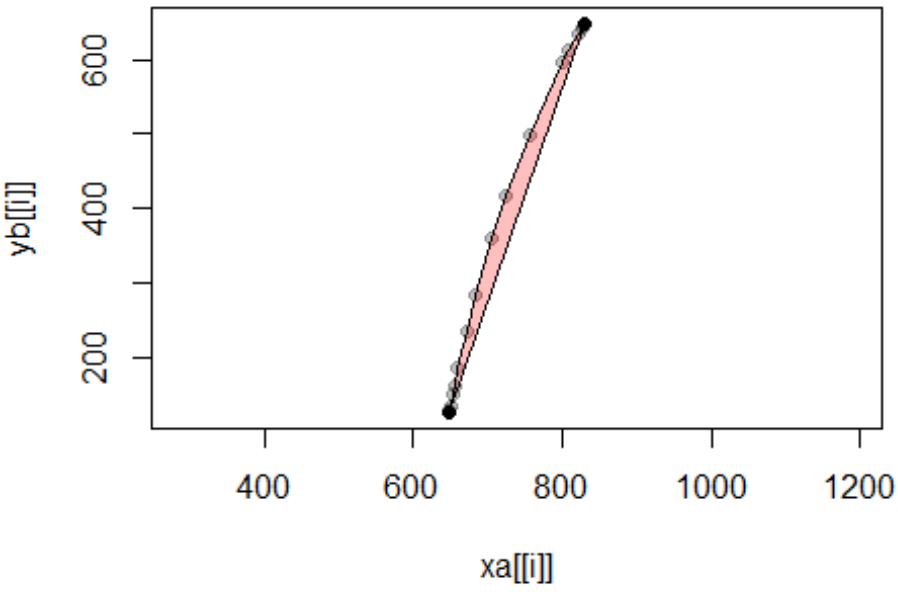

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

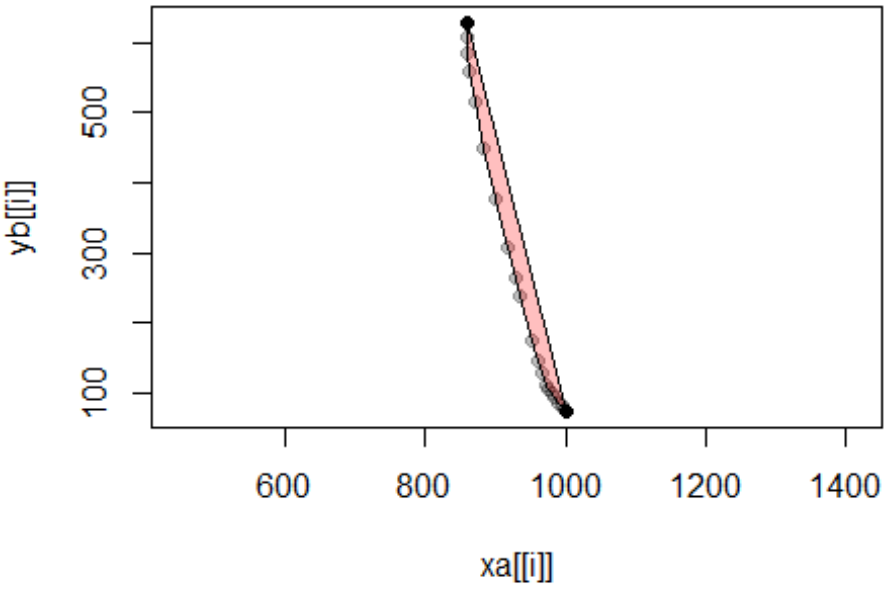

yb[[i]]
200
300
400
500
600
400
600
800
1000
xa[[i]]

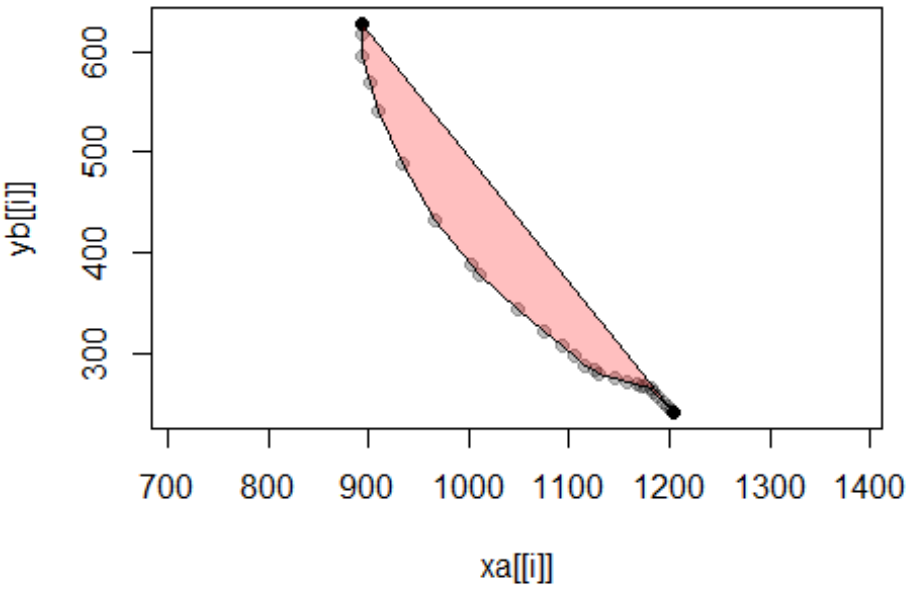

yb[[i]]
600
400
200
600
800
1000
1200
xa[[i]]

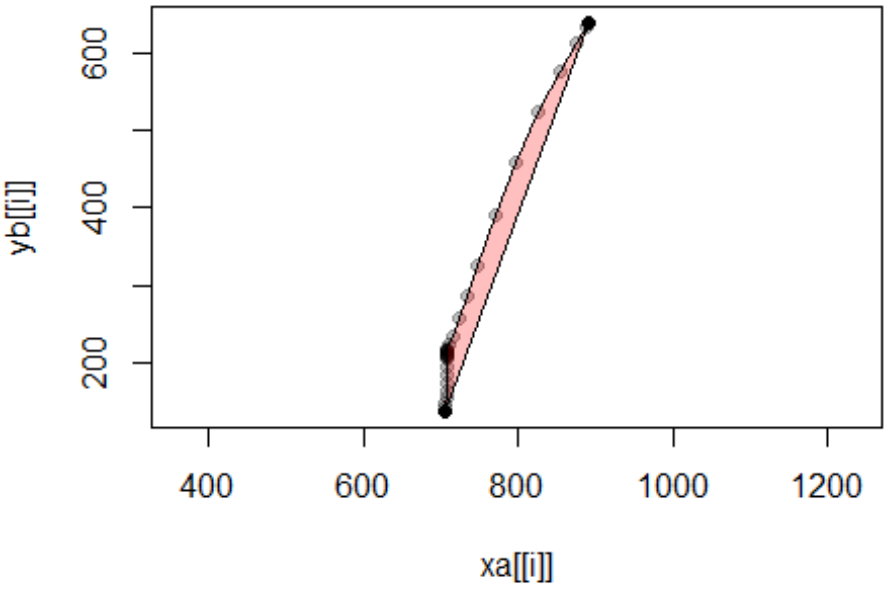

yb[[i]]
600
500
400
300
200
400
600
800
1000
1200
xa[[i]]

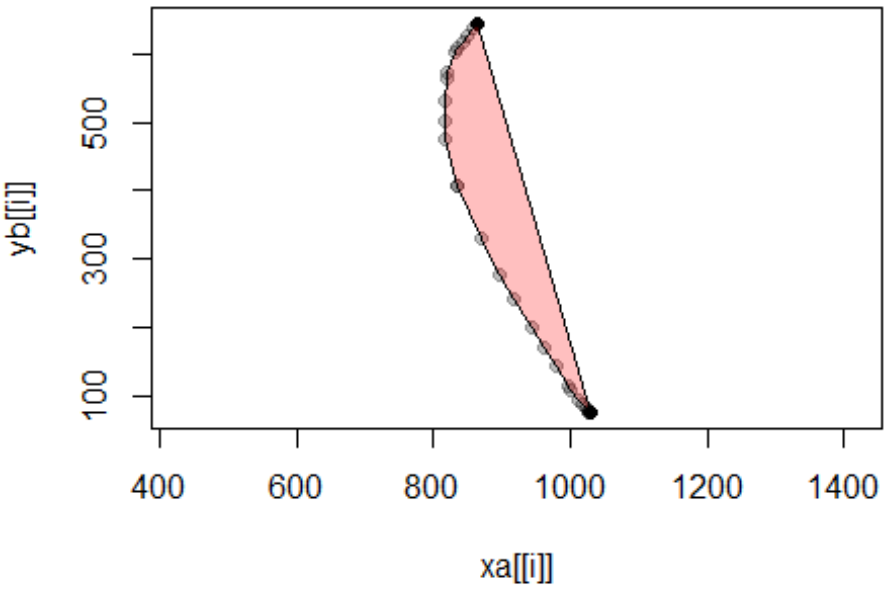

yb[[i]]
200
400
600
200
400
600
800
1000
xa[[i]]

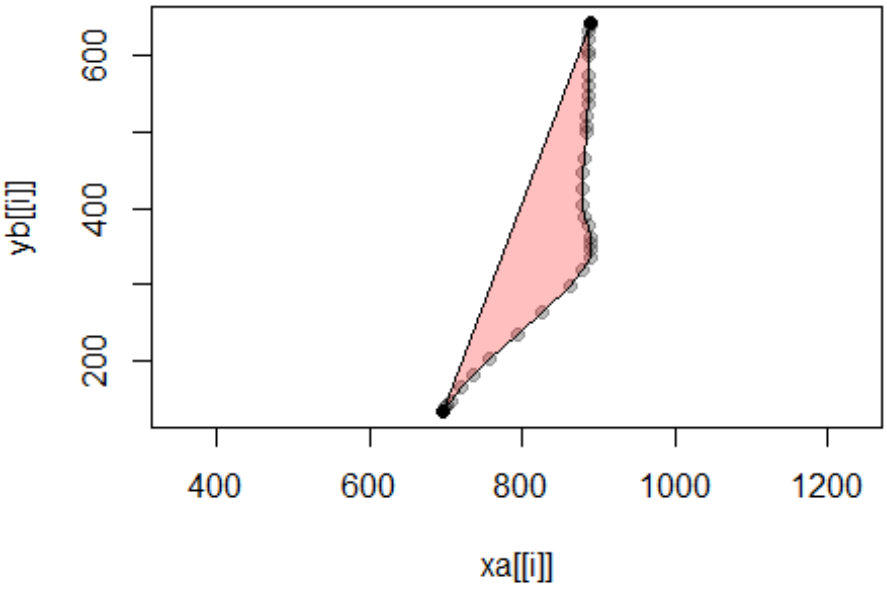

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

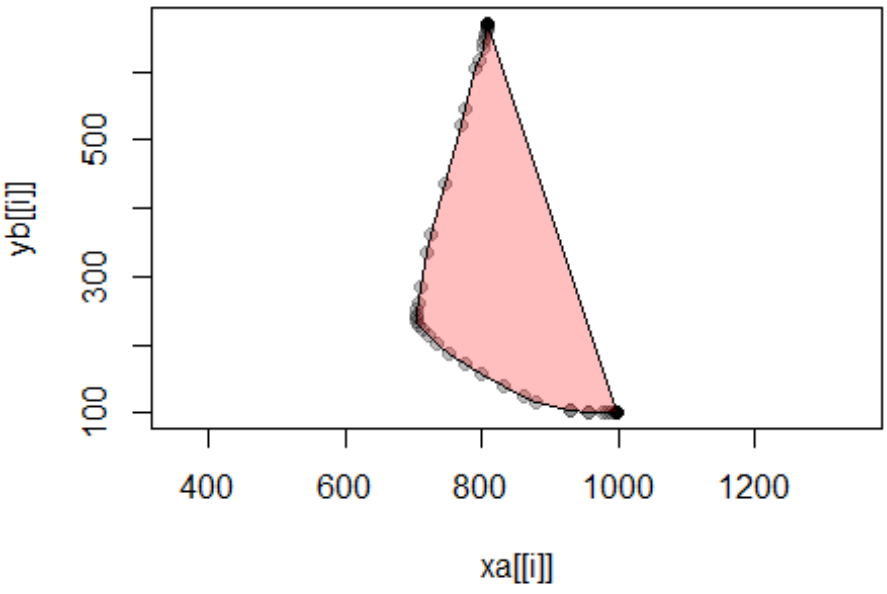

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

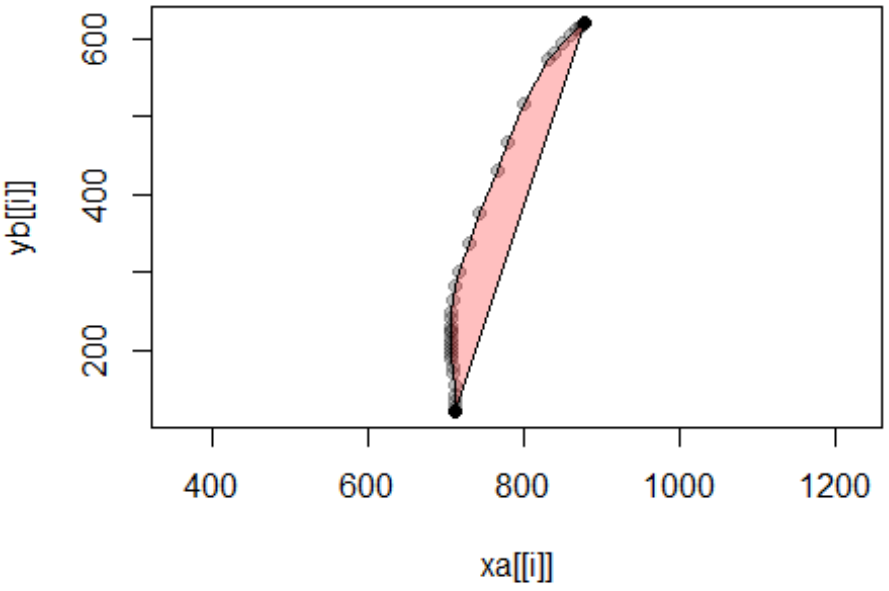

yb[[i]]
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
xa[[i]]

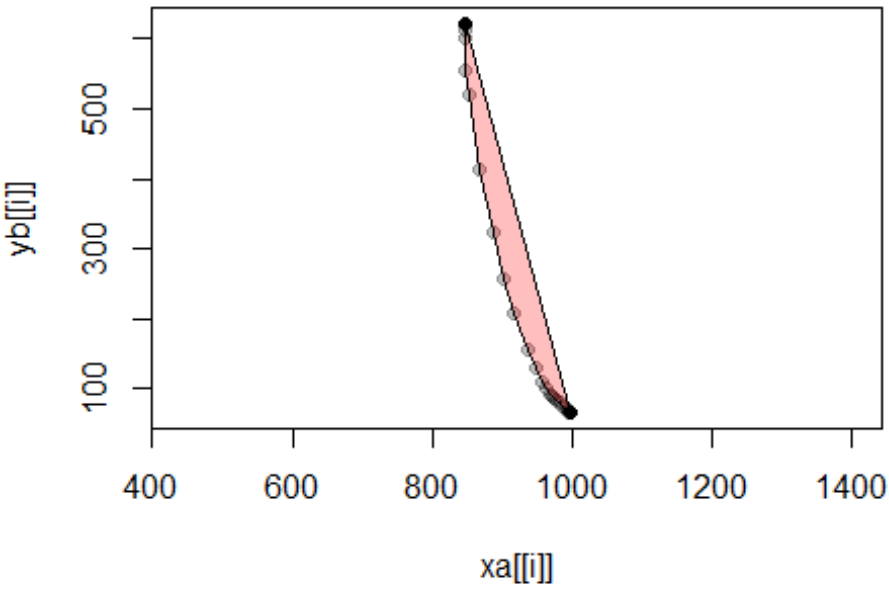

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

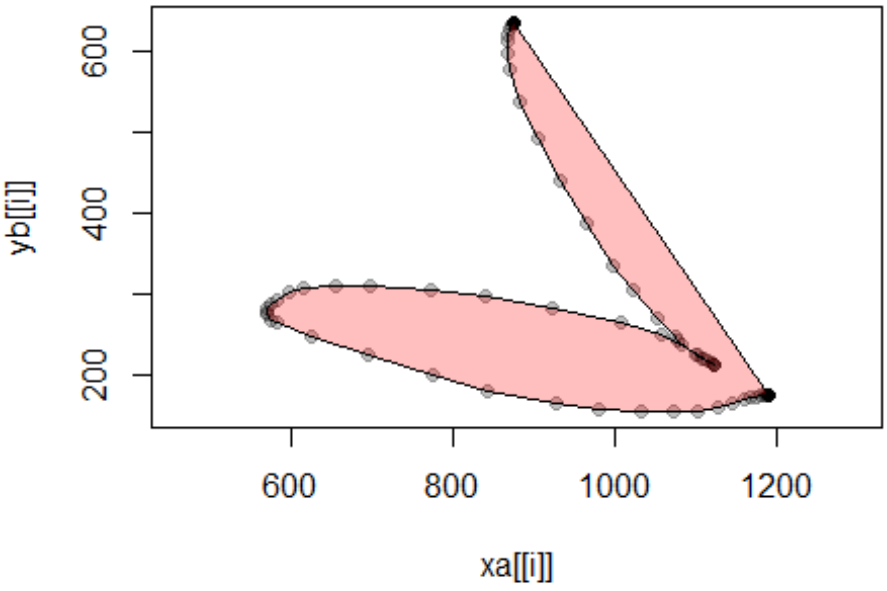

yb[[i]]
100
300
500
400
600
800
1000
1200
1400
xa[[i]]

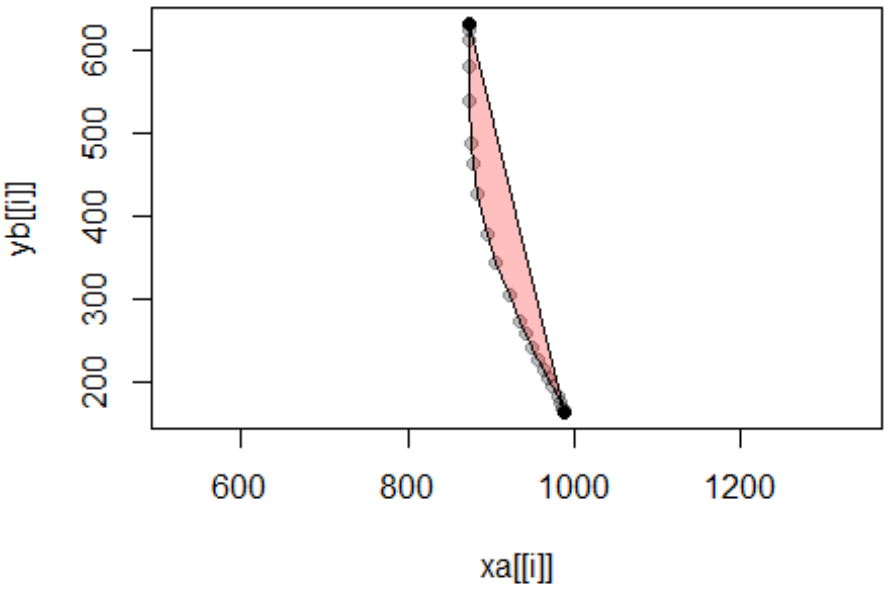

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

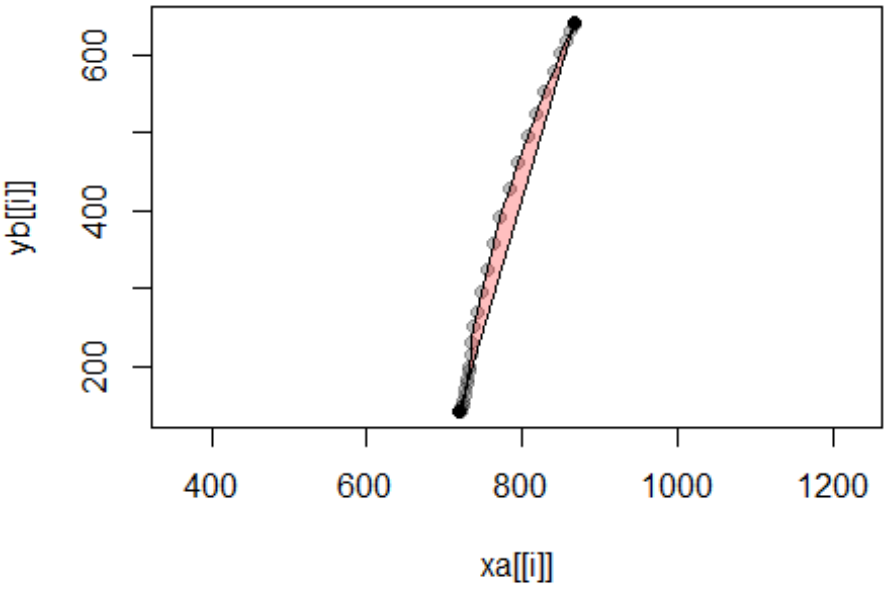

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

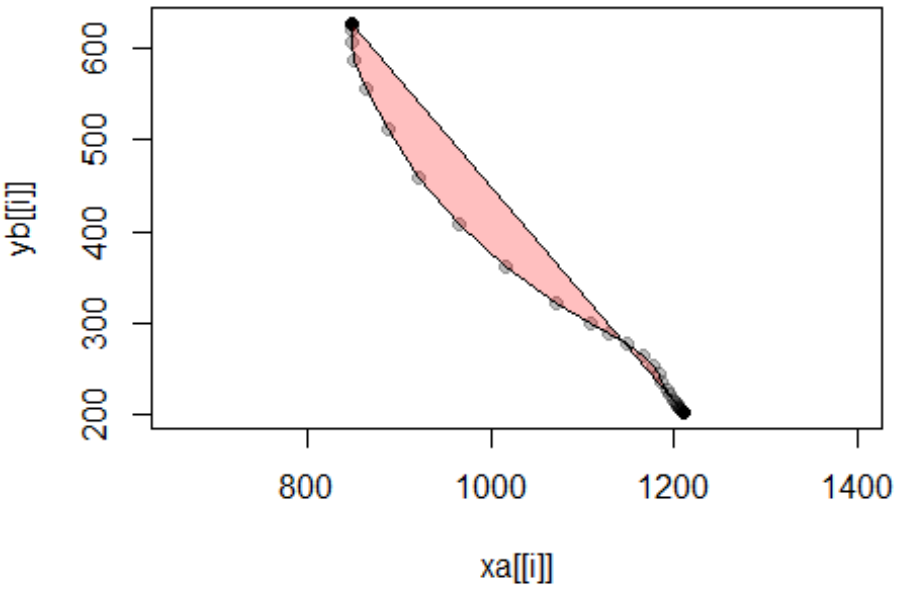

yb[[i]]
600
400
200
400
600
800
1000
1200
xa[[i]]

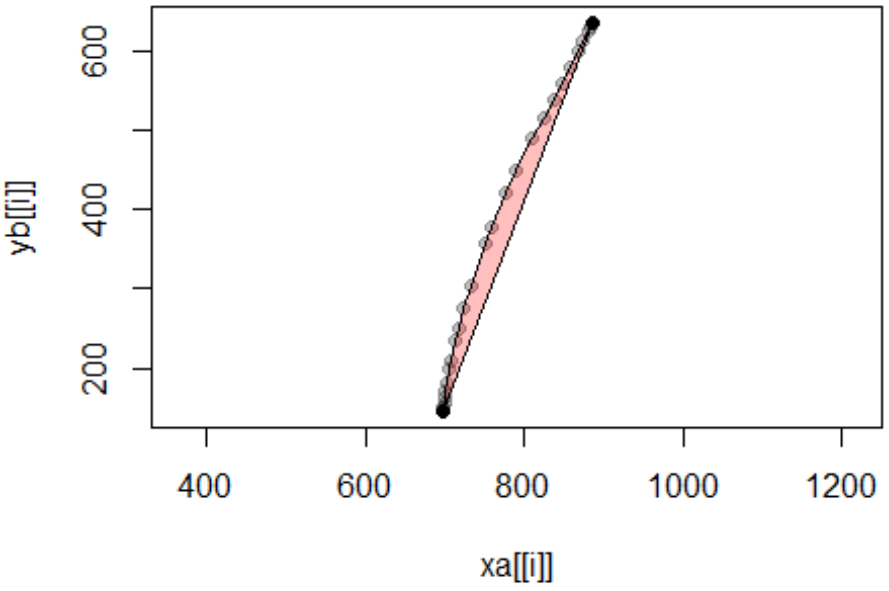

yb[[i]]
500
300
100
400
600
800
1000
1200
1400
xa[[i]]

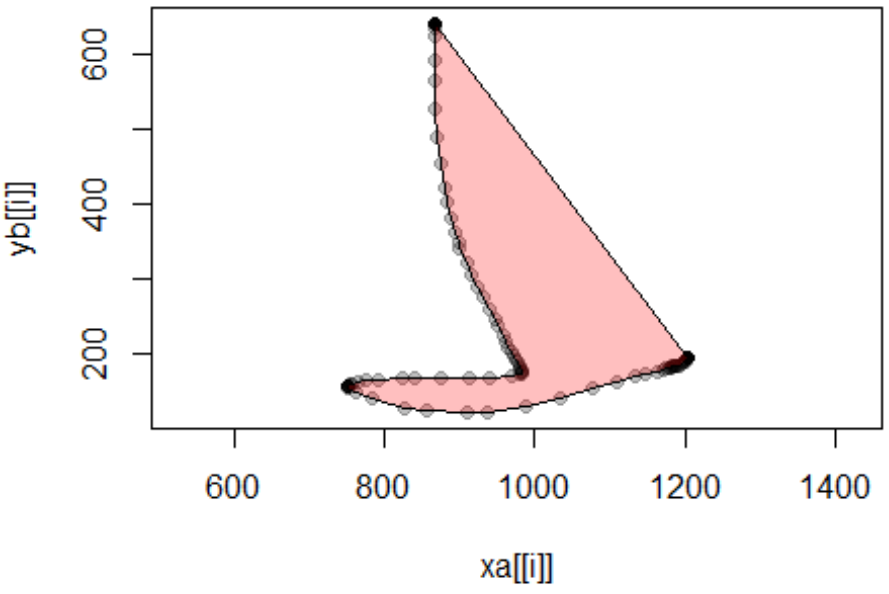

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

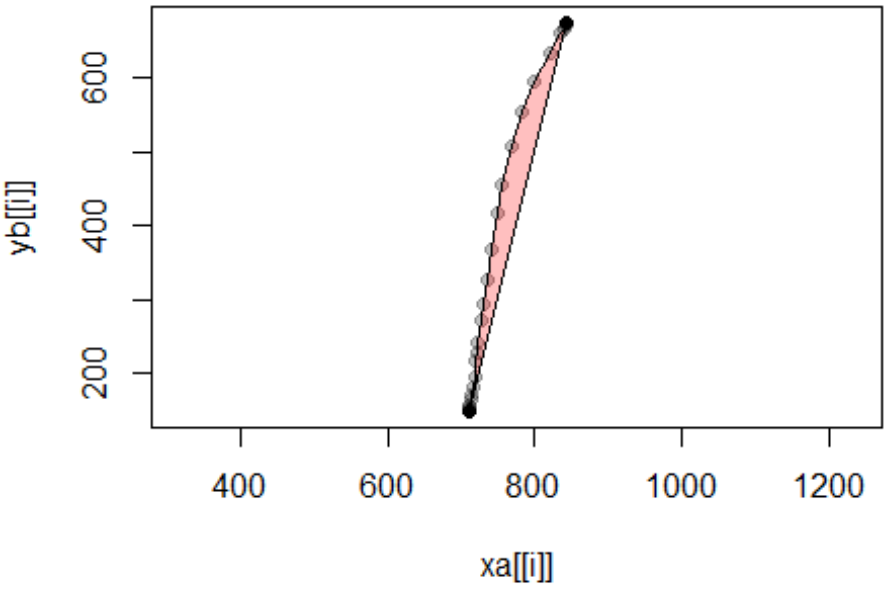

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

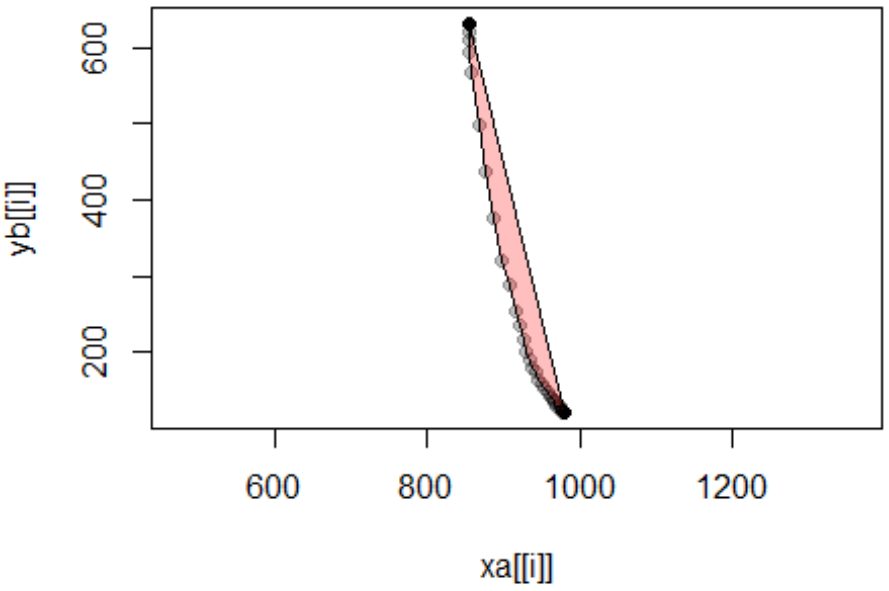

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

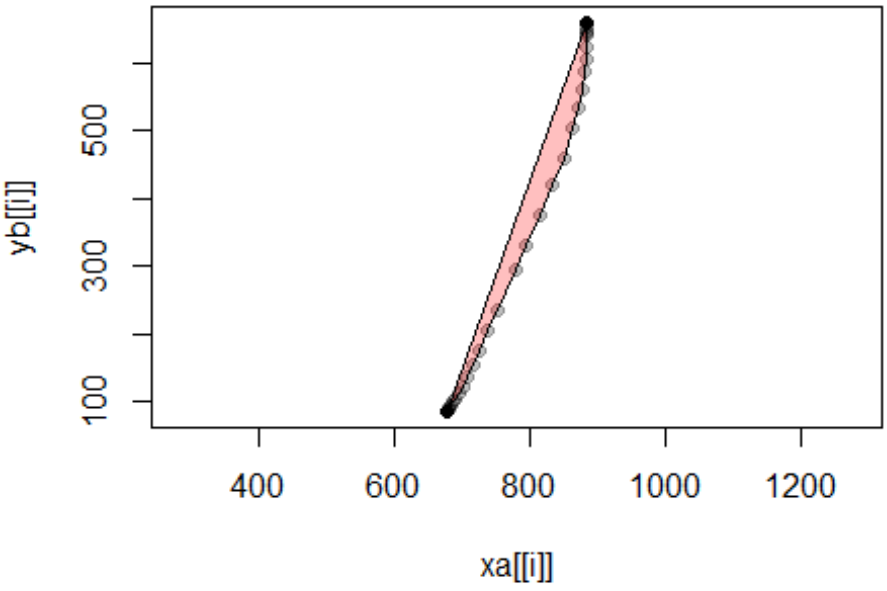

yb[[i]]
200
300
400
500
600
800
1000
1200
1400
xa[[i]]

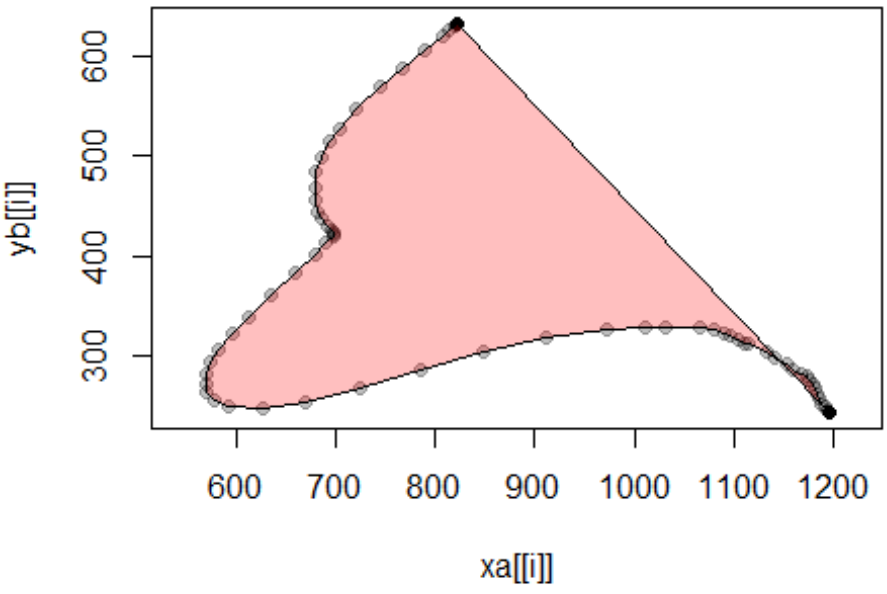

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

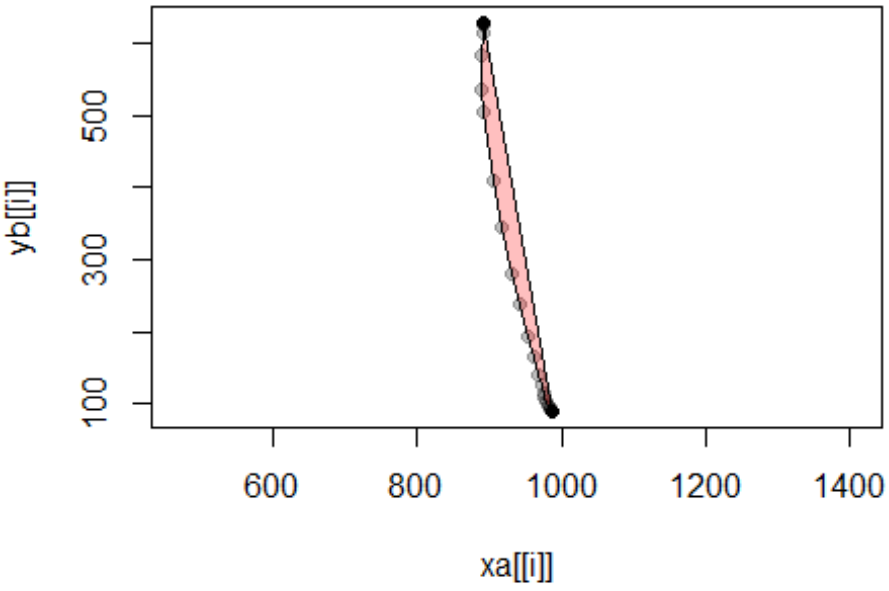

yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]

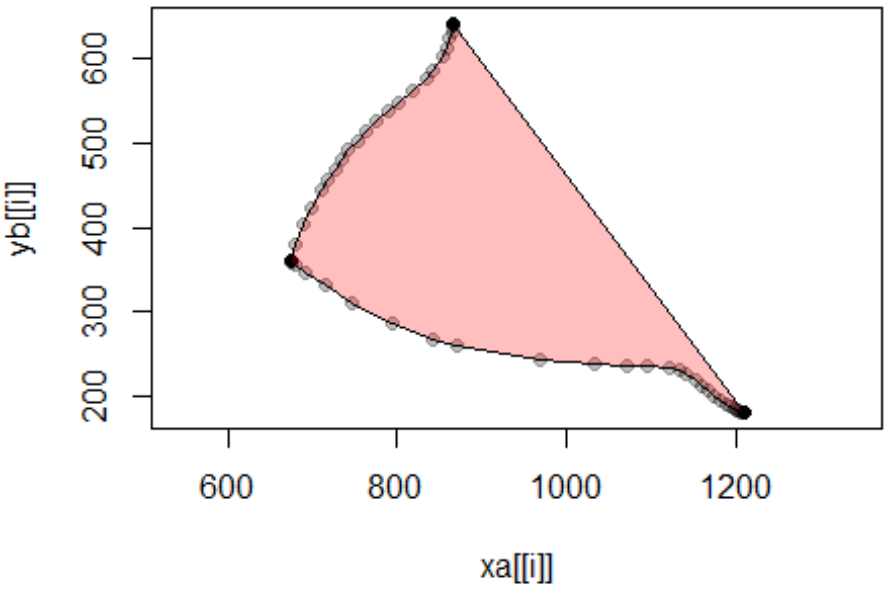

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]

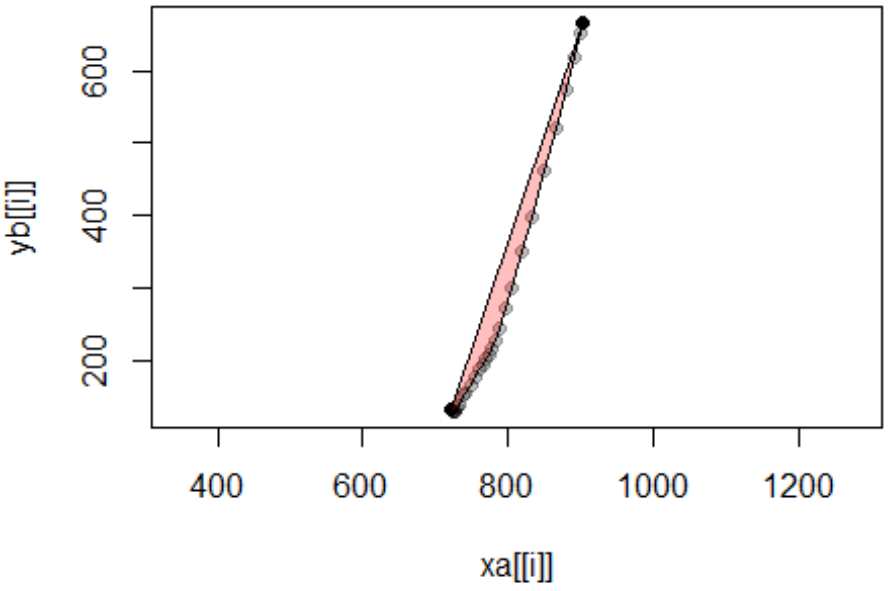

yb[[i]]
200
400
600
600
800
1000
1200
xa[[i]]

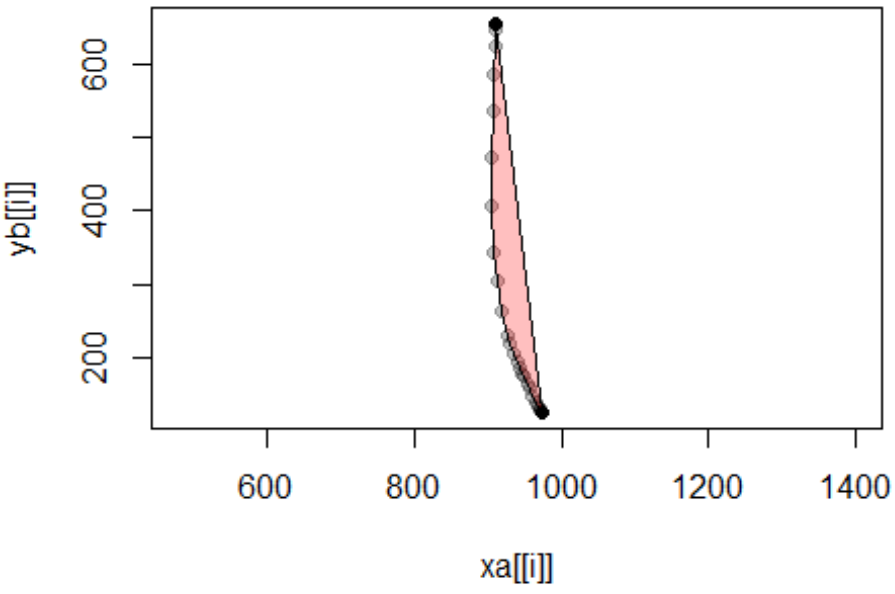

yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]

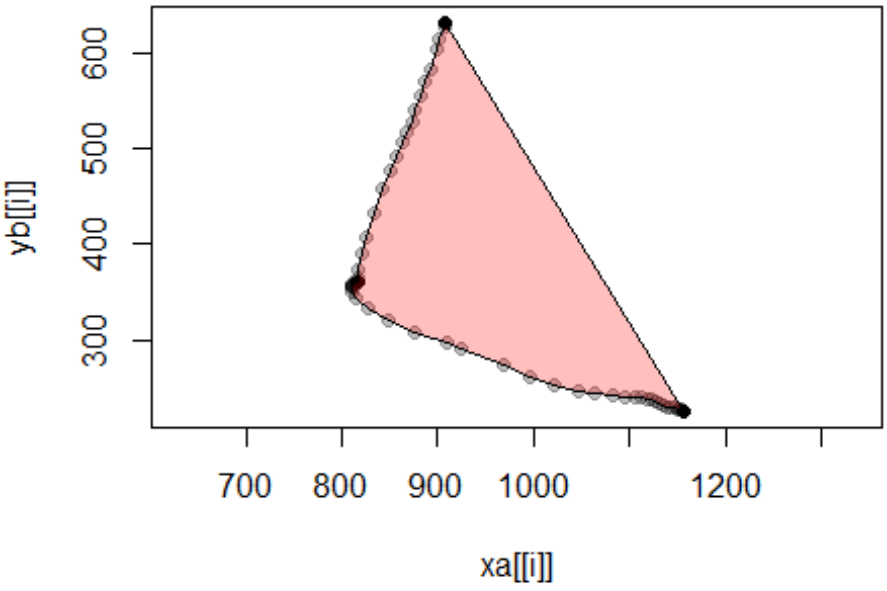

yb[[i]]
300
400
500
600
600
700
800
900
1000
1100
1200
xa[[i]]

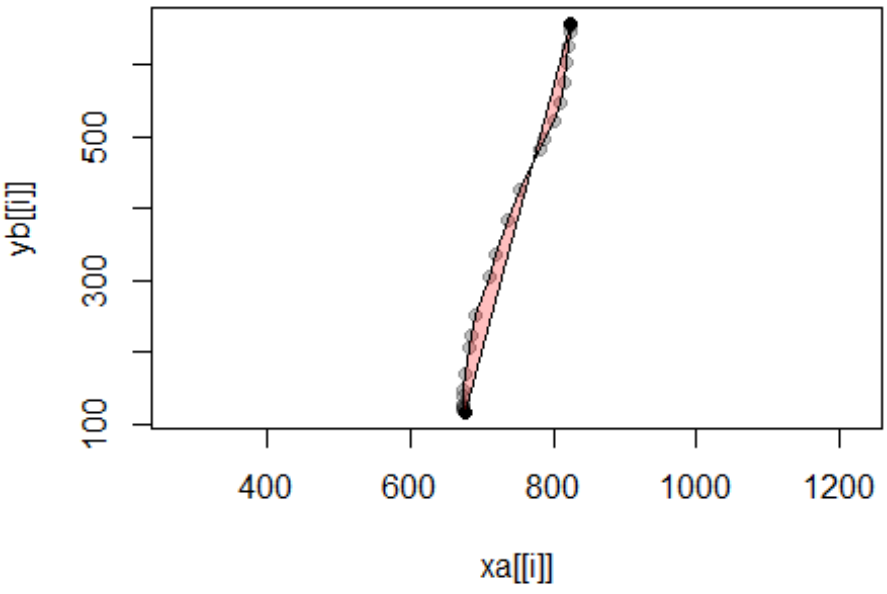

yb[[i]]
100
300
500
600
800
1000
1200
1400
xa[[i]]

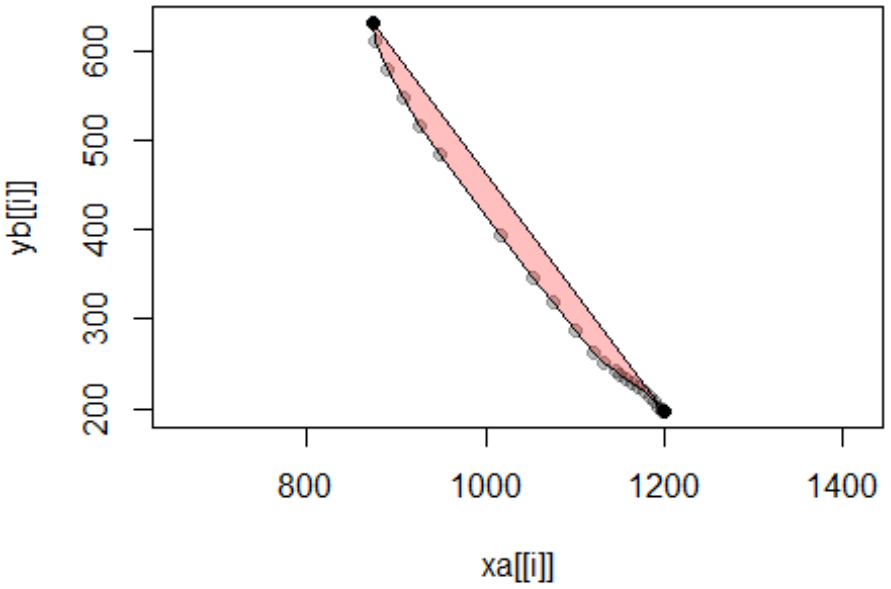

yb[[i]]
200
300
400
500
600
600
800
1000
1200
xa[[i]]

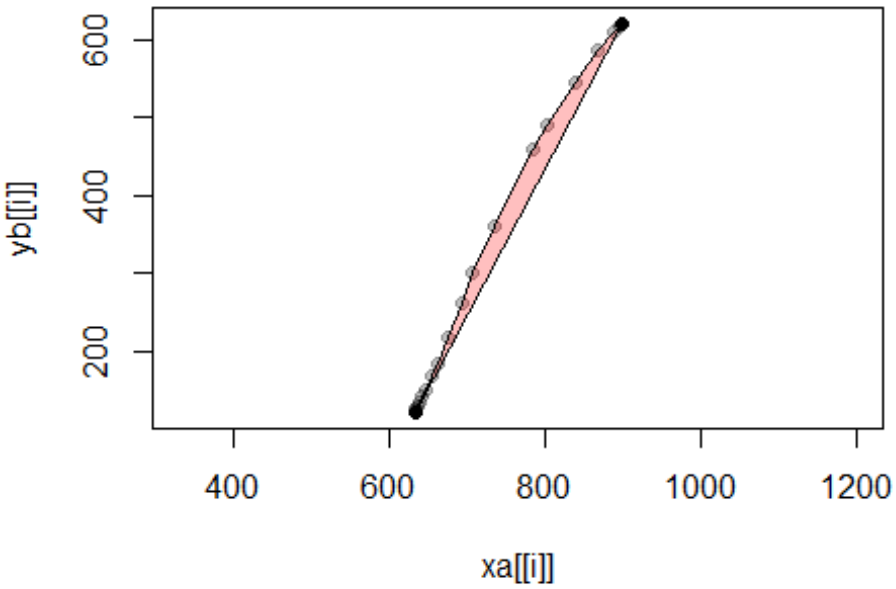

yb[[i]]
200
400
600
400
600
800
1000
1200
xa[[i]]


yb[[i]]
200
400
600
600
800
1000
1200
1400
xa[[i]]


yb[[i]]
300
400
500
600
700
800
900
1000
1200
xa[[i]]


yb[[i]]
100
300
500
400
600
800
1000
1200
xa[[i]]






```
##########################################################

## créé après avoir rempli le vecteur AUC
dataframe = data.frame(subject, frequency, congruency, MT, AUC, IT, groupe,
```

```
tousPart$essais.couleur, tousPart$essais.mot)
################################################
#MC/MI
for (i in 1:nrow(dataframe)) {


if(dataframe$congruency[i] == "0"){

  congruency[i] <- c("I")
}
else if (dataframe$congruency[i] == "1"){

  congruency[i] <- c("C")
}
}


 for (i in 1:nrow(dataframe)) {

    if(dataframe$groupe[i] == "1" && dataframe$tousPart.essais.couleur[i] ==
"blue" && dataframe$tousPart.essais.mot[i] == "BLEU" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "1" &&
dataframe$tousPart.essais.couleur[i] == "yellow" &&
dataframe$tousPart.essais.mot[i] == "JAUNE" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "1" &&
dataframe$tousPart.essais.couleur[i] == "yellow" &&
dataframe$tousPart.essais.mot[i] == "BLEU" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "1" &&
dataframe$tousPart.essais.couleur[i] == "blue" &&
dataframe$tousPart.essais.mot[i] == "JAUNE" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "1" &&
dataframe$tousPart.essais.couleur[i] == "green" &&
dataframe$tousPart.essais.mot[i] == "ROUGE" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "1" &&
dataframe$tousPart.essais.couleur[i] == "green" &&
dataframe$tousPart.essais.mot[i] == "VERT" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "1" &&
```

```
dataframe$tousPart.essais.couleur[i] == "red" &&
dataframe$tousPart.essais.mot[i] == "ROUGE" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "1" &&
dataframe$tousPart.essais.couleur[i] == "red" &&
dataframe$tousPart.essais.mot[i] == "VERT" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "blue" &&
dataframe$tousPart.essais.mot[i] == "BLEU" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "yellow" &&
dataframe$tousPart.essais.mot[i] == "JAUNE" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "yellow" &&
dataframe$tousPart.essais.mot[i] == "BLEU" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "blue" &&
dataframe$tousPart.essais.mot[i] == "JAUNE" ){
      frequency[i] <- c("MI")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "green" &&
dataframe$tousPart.essais.mot[i] == "ROUGE" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "green" &&
dataframe$tousPart.essais.mot[i] == "VERT" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "red" &&
dataframe$tousPart.essais.mot[i] == "ROUGE" ){
      frequency[i] <- c("MC")
    }
    else if(dataframe$groupe[i] == "2" &&
dataframe$tousPart.essais.couleur[i] == "red" &&
dataframe$tousPart.essais.mot[i] == "VERT" ){
      frequency[i] <- c("MC")
    }
  }
```

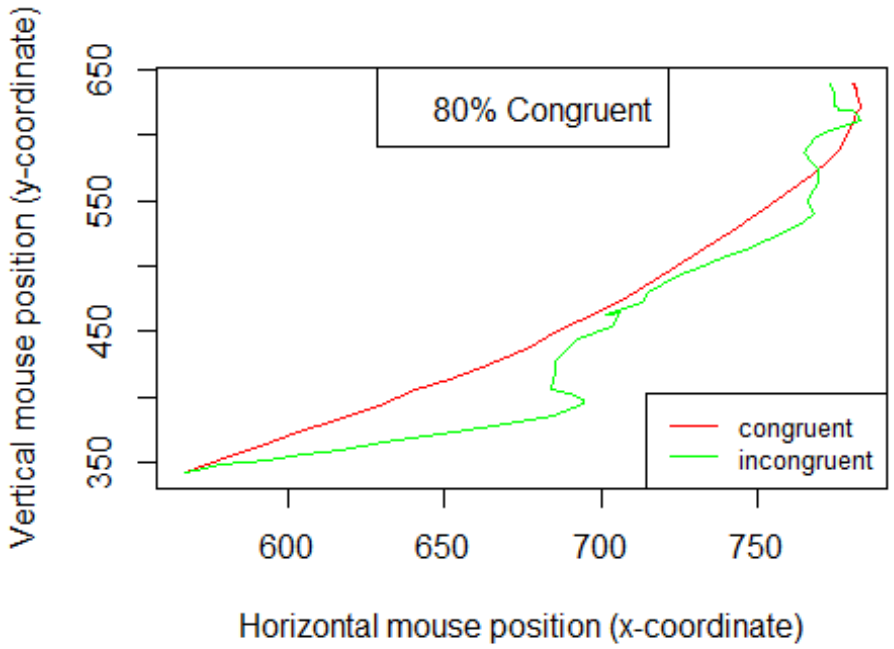
```r
dataframe = data.frame(subject, frequency, congruency, MT, AUC, IT, groupe)

############################

############################
#####moyenne des courbes MC vs MI


cxmcc <- cxmci <- cxmic <- cxmii <- list()

cymcc <- cymci <- cymic <- cymii <- list()

##### x coordinates gp1

for (i in 1:nrow(tousPart)) {
  if(tousPart$essais.mot[i] == "BLEU" && tousPart$essais.couleur[i] == "blue"
&& tousPart$groupe[i] == "1"){
    cxmcc[i] <- tousPart$essais.courbeX[i]
  }
  else if(tousPart$essais.mot[i] == "JAUNE" && tousPart$essais.couleur[i] ==
"blue" && tousPart$groupe[i] == "1"){
    cxmci[i] <- tousPart$essais.courbeX[i]
  }
  else if(tousPart$essais.mot[i] == "VERT" && tousPart$essais.couleur[i] ==
"green" && tousPart$groupe[i] == "1"){
    cxmic[i] <- tousPart$essais.courbeX[i]
  }
  else if(tousPart$essais.mot[i] == "ROUGE" && tousPart$essais.couleur[i] ==
"green" && tousPart$groupe[i] == "1"){
    cxmii[i] <- tousPart$essais.courbeX[i]
  }
}
cxmcc = cxmcc[-which(sapply(cxmcc, is.null))]
cxmci = cxmci[-which(sapply(cxmci, is.null))]
cxmic = cxmic[-which(sapply(cxmic, is.null))]
cxmii = cxmii[-which(sapply(cxmii, is.null))]

##### Y coordinates gp1

for (i in 1:nrow(tousPart)) {
  if(tousPart$essais.mot[i] == "BLEU" && tousPart$essais.couleur[i] == "blue"
&& tousPart$groupe[i] == "1"){
    cymcc[i] <- tousPart$essais.courbeY[i]
  }
  else if(tousPart$essais.mot[i] == "JAUNE" && tousPart$essais.couleur[i] ==
"blue" && tousPart$groupe[i] == "1"){
    cymci[i] <- tousPart$essais.courbeY[i]
  }
```

```
  else if(tousPart$essais.mot[i] == "VERT" && tousPart$essais.couleur[i] ==
"green" && tousPart$groupe[i] == "1"){
    cymic[i] <- tousPart$essais.courbeY[i]
  }
  else if(tousPart$essais.mot[i] == "ROUGE" && tousPart$essais.couleur[i] ==
"green" && tousPart$groupe[i] == "1"){
    cymii[i] <- tousPart$essais.courbeY[i]
  }
}
cymcc = cymcc[-which(sapply(cymcc, is.null))]
cymci = cymci[-which(sapply(cymci, is.null))]
cymic = cymic[-which(sapply(cymic, is.null))]
cymii = cymii[-which(sapply(cymii, is.null))]


########################################

## X & Y coordinates gp2


cxmcc2 <- cxmci2 <- cxmic2 <- cxmii2 <- list()

cymcc2 <- cymci2 <- cymic2 <- cymii2 <- list()

##### x coordinates gp2

for (i in 1:nrow(tousPart)) {
  if(tousPart$essais.mot[i] == "ROUGE" && tousPart$essais.couleur[i] == "red"
&& tousPart$groupe[i] == "2"){
    cxmcc2[i] <- tousPart$essais.courbeX[i]
  }
  else if(tousPart$essais.mot[i] == "VERT" && tousPart$essais.couleur[i] ==
"red" && tousPart$groupe[i] == "2"){
    cxmci2[i] <- tousPart$essais.courbeX[i]
  }
  else if(tousPart$essais.mot[i] == "JAUNE" && tousPart$essais.couleur[i] ==
"yellow" && tousPart$groupe[i] == "2"){
    cxmic2[i] <- tousPart$essais.courbeX[i]
  }
  else if(tousPart$essais.mot[i] == "BLEU" && tousPart$essais.couleur[i] ==
"yellow" && tousPart$groupe[i] == "2"){
    cxmii2[i] <- tousPart$essais.courbeX[i]
  }
}
cxmcc2 = cxmcc2[-which(sapply(cxmcc2, is.null))]
cxmci2 = cxmci2[-which(sapply(cxmci2, is.null))]
cxmic2 = cxmic2[-which(sapply(cxmic2, is.null))]
cxmii2 = cxmii2[-which(sapply(cxmii2, is.null))]
```

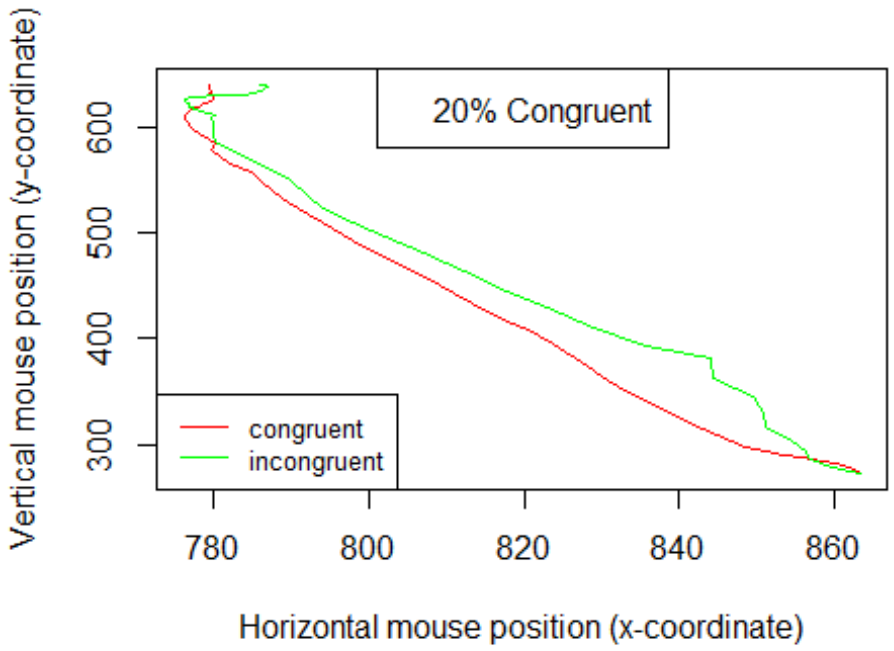
```
##### Y coordinates gp2

for (i in 1:nrow(tousPart)) {
  if(tousPart$essais.mot[i] == "ROUGE" && tousPart$essais.couleur[i] == "red"
&& tousPart$groupe[i] == "2"){
    cymcc2[i] <- tousPart$essais.courbeY[i]
  }
  else if(tousPart$essais.mot[i] == "VERT" && tousPart$essais.couleur[i] ==
"red" && tousPart$groupe[i] == "2"){
    cymci2[i] <- tousPart$essais.courbeY[i]
  }
  else if(tousPart$essais.mot[i] == "JAUNE" && tousPart$essais.couleur[i] ==
"yellow" && tousPart$groupe[i] == "2"){
    cymic2[i] <- tousPart$essais.courbeY[i]
  }
  else if(tousPart$essais.mot[i] == "BLEU" && tousPart$essais.couleur[i] ==
"yellow" && tousPart$groupe[i] == "2"){
    cymii2[i] <- tousPart$essais.courbeY[i]
  }
}
cymcc2 = cymcc2[-which(sapply(cymcc2, is.null))]
cymci2 = cymci2[-which(sapply(cymci2, is.null))]
cymic2 = cymic2[-which(sapply(cymic2, is.null))]
cymii2 = cymii2[-which(sapply(cymii2, is.null))]

#####################################################################


####plot for MC c*I   gp2

mcc2 <- data.frame( cbind(cxmcc2, cymcc2))
mci2 <- data.frame( cbind(cxmci2, cymci2))

a <- b <- xcc2 <- ycc2 <- c()

for (j in 1:150) {
  for (i in 1:nrow(mcc2)) {
    a[i] <- mcc2$cxmcc2[[i]][j]
    b[i] <- mcc2$cymcc2[[i]][j]
  }
  xcc2[j] <- mean(a)
  ycc2[j] <- mean(b)
}

xcc2 <- na.omit(xcc2)
ycc2 <- na.omit(ycc2)


a <- b <- xci2 <- yci2 <- c()
```

```r
for (j in 1:150) {
  for (i in 1:nrow(mci2)) {
    a[i] <- mci2$cxmci2[[i]][j]
    b[i] <- mci2$cymci2[[i]][j]
  }
  xci2[j] <- mean(a)
  yci2[j] <- mean(b)
}


xci2 <- na.omit(xci2)
yci2 <- na.omit(yci2)


####################################3

#plot for Mi c*I  gp2


mic2 <- data.frame( cbind(cxmic2, cymic2))

a <- b <- xic2 <- yic2 <- c()

for (j in 1:150) {
  for (i in 1:nrow(mic2)) {
    a[i] <- mic2$cxmic2[[i]][j]
    b[i] <- mic2$cymic2[[i]][j]
  }
  xic2[j] <- mean(a)
  yic2[j] <- mean(b)
}

xic2 <- na.omit(xic2)
yic2 <- na.omit(yic2)

mii2 <- data.frame( cbind(cxmii2, cymii2))

a <- b <- xii2 <- yii2 <- c()

for (j in 1:150) {
  for (i in 1:nrow(mii2)) {
    a[i] <- mii2$cxmii[[i]][j]
    b[i] <- mii2$cymii[[i]][j]
  }
  xii2[j] <- mean(a)
  yii2[j] <- mean(b)
}
```

```r
xii2 <- na.omit(xii2)
yii2 <- na.omit(yii2)



##################

plot( xcc2, ycc2, type="l", col="red", ylab = "Vertical mouse position (y-
coordinate)", xlab = "Horizontal mouse position (x-coordinate)")
legend("bottomright", 95, legend=c("congruent", "incongruent"),
       col=c("red", "green"), lty=1:1, cex=0.8)
legend("top", 95, legend=c("80% Congruent"))
par(new=TRUE)
plot( xci2, yci2, type="l", col="green", axes=FALSE,ann=FALSE, )
```



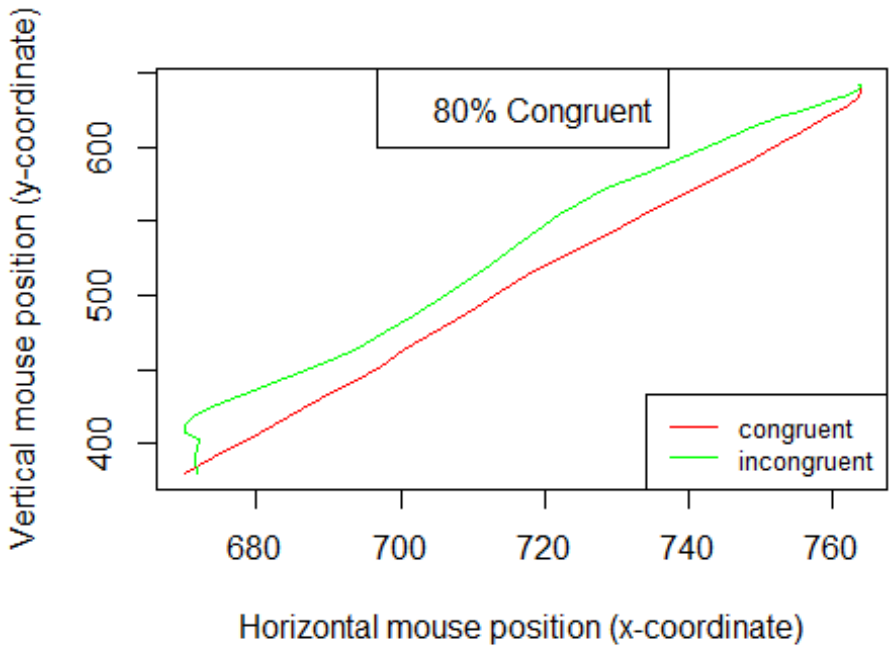

```r
################
plot( xic2, yic2, type="l", col="red", ylab = "Vertical mouse position (y-
coordinate)", xlab = "Horizontal mouse position (x-coordinate)")
legend("bottomleft", 95, legend=c("congruent", "incongruent"),
       col=c("red", "green"), lty=1:1, cex=0.8)
legend("top", 95, legend=c("20% Congruent"))
par(new=TRUE)
plot( xii2, yii2, type="l", col="green", axes=FALSE,ann=FALSE, )
```



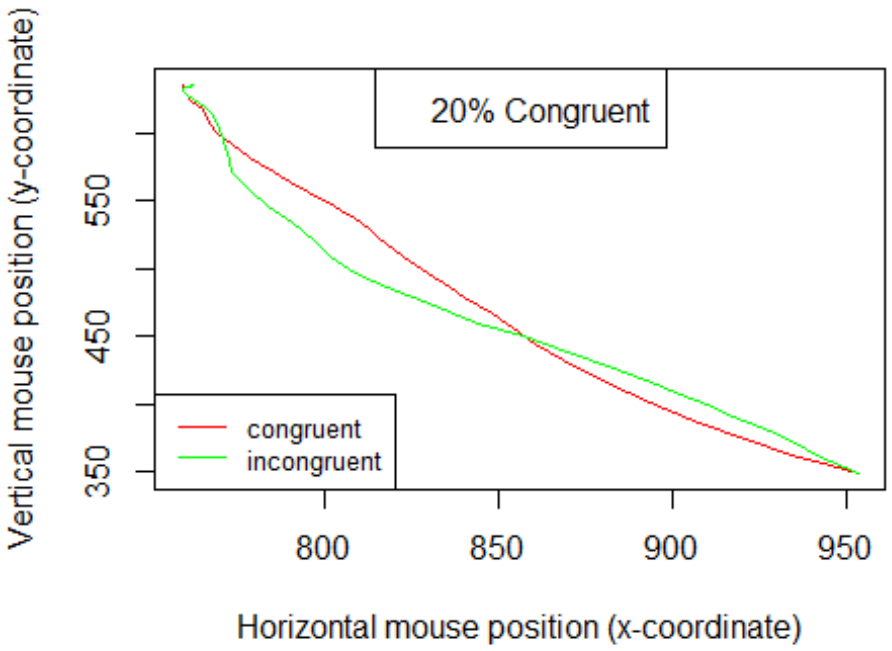

```
##########################################

####plot for MC c*I   gp1

mcc <- data.frame( cbind(cxmcc, cymcc))
mci <- data.frame( cbind(cxmci, cymci))

a <- b <- xcc <- ycc <- c()

for (j in 1:150) {
  for (i in 1:nrow(mcc)) {
    a[i] <- mcc$cxmcc[[i]][j]
    b[i] <- mcc$cymcc[[i]][j]
  }
  xcc[j] <- mean(a)
  ycc[j] <- mean(b)
}

xcc <- na.omit(xcc)
ycc <- na.omit(ycc)


a <- b <- xci <- yci <- c()

for (j in 1:150) {
  for (i in 1:nrow(mci)) {
```

```
    a[i] <- mci$cxmci[[i]][j]
    b[i] <- mci$cymci[[i]][j]
  }
  xci[j] <- mean(a)
  yci[j] <- mean(b)
}


xci <- na.omit(xci)
yci <- na.omit(yci)


####################################3

#plot for Mi c*I  gp1


mic <- data.frame( cbind(cxmic, cymic))

a <- b <- xic <- yic <- c()

for (j in 1:150) {
  for (i in 1:nrow(mic)) {
    a[i] <- mic$cxmic[[i]][j]
    b[i] <- mic$cymic[[i]][j]
  }
  xic[j] <- mean(a)
  yic[j] <- mean(b)
}

xic <- na.omit(xic)
yic <- na.omit(yic)

mii <- data.frame( cbind(cxmii, cymii))

a <- b <- xii <- yii <- c()

for (j in 1:150) {
  for (i in 1:nrow(mii)) {
    a[i] <- mii$cxmii[[i]][j]
    b[i] <- mii$cymii[[i]][j]
  }
  xii[j] <- mean(a)
  yii[j] <- mean(b)
}

xii <- na.omit(xii)
yii <- na.omit(yii)
```

```
#################

plot( xcc, ycc, type="l", col="red", ylab = "Vertical mouse position (y-
coordinate)", xlab = "Horizontal mouse position (x-coordinate)")
legend("bottomright", 95, legend=c("congruent", "incongruent"),
       col=c("red", "green"), lty=1:1, cex=0.8)
legend("top", 95, legend=c("80% Congruent"))
par(new=TRUE)
plot( xci, yci, type="l", col="green", axes=FALSE,ann=FALSE, )
```



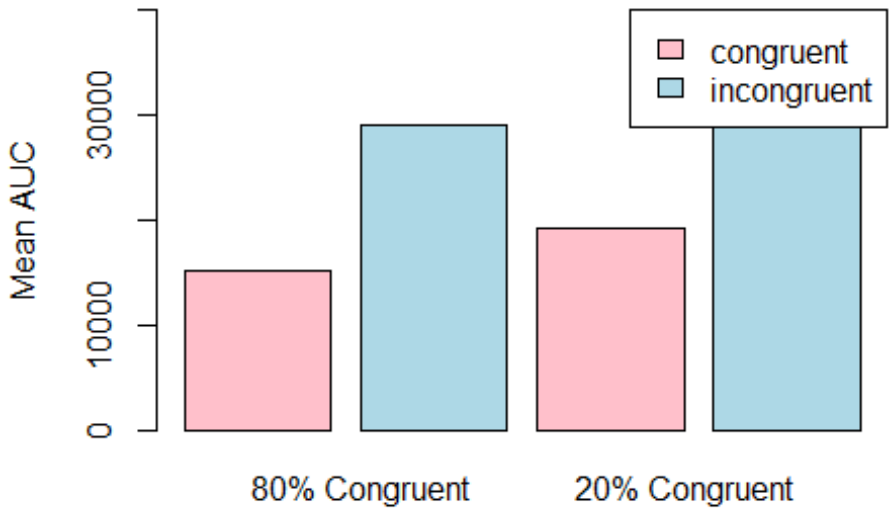

```
################
plot( xic, yic, type="l", col="red", ylab = "Vertical mouse position (y-
coordinate)", xlab = "Horizontal mouse position (x-coordinate)")
legend("bottomleft", 95, legend=c("congruent", "incongruent"),
       col=c("red", "green"), lty=1:1, cex=0.8)
legend("top", 95, legend=c("20% Congruent"))
par(new=TRUE)
plot( xii, yii, type="l", col="green", axes=FALSE,ann=FALSE, )
```



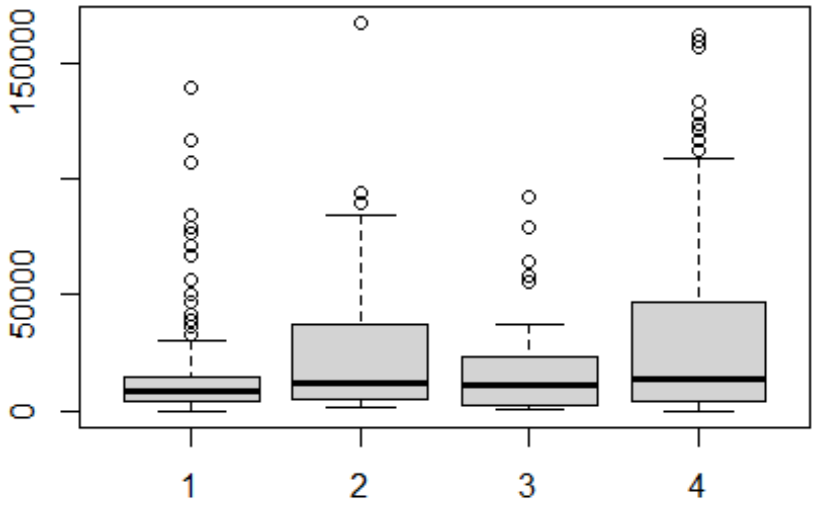

```
##############################################################################

##############################################################################
## Analyse des donnees


## AUC et MT pour groupe 1

freq <- gp <- auc <- cong <- mt <-it <- c()

     for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "C" && dataframe$frequency[i] == "MC" &&
dataframe$groupe[i] == "1"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]


  }
     }
```

```
dat.MCC <- data.frame (gp, cong, freq, auc, mt, it)
dat.MCC <- na.omit(dat.MCC)

freq <- gp <- auc <- cong <- mt <-it <- c()
for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "I" && dataframe$frequency[i] == "MC" &&
dataframe$groupe[i] == "1"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MCI <- data.frame (gp, cong, freq, auc, mt, it)
dat.MCI <- na.omit(dat.MCI)

freq <- gp <- auc <- cong <- mt <-it <- c()
for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "C" && dataframe$frequency[i] == "MI" &&
dataframe$groupe[i] == "1"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MIC <- data.frame (gp, cong, freq, auc, mt, it)
dat.MIC <- na.omit(dat.MIC)

freq <- gp <- auc <- cong <- mt <-it <- c()
for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "I" && dataframe$frequency[i] == "MI" &&
dataframe$groupe[i] == "1"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
```

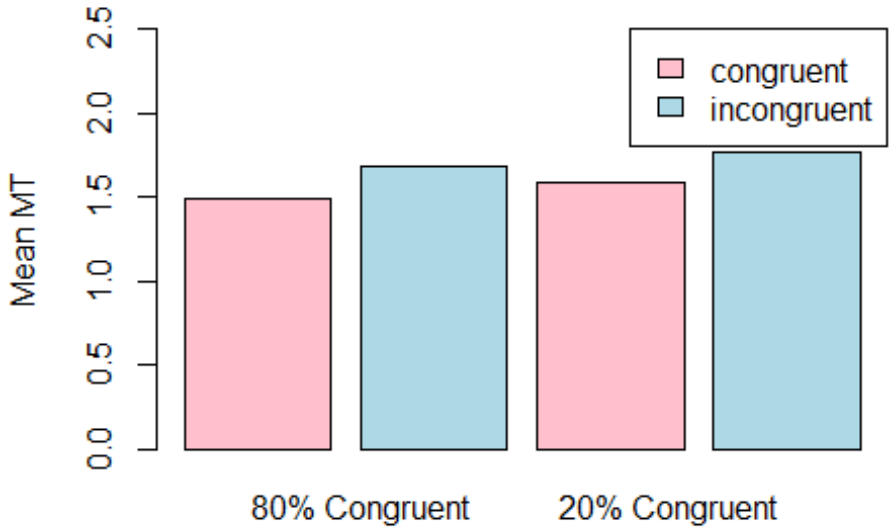
```
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MII <- data.frame (gp, cong, freq, auc, mt, it)
dat.MII <- na.omit(dat.MII)

lauc <- as.vector( rbind(
mean(dat.MCC$auc),mean(dat.MCI$auc),mean(dat.MIC$auc), mean(dat.MII$auc)
)[,1] )

barplot(lauc, ylim = c(0,40000), col=c("pink","lightblue"), xaxt="n",ylab =
"Mean AUC",legend.text = c("congruent","incongruent"),
        args.legend = list(x = "topright"))
axis(1, at = c(1.4, 0, 3.6, 0), labels = c("80% Congruent","","20%
Congruent", "") ,tick=FALSE, cex=0.5)
```



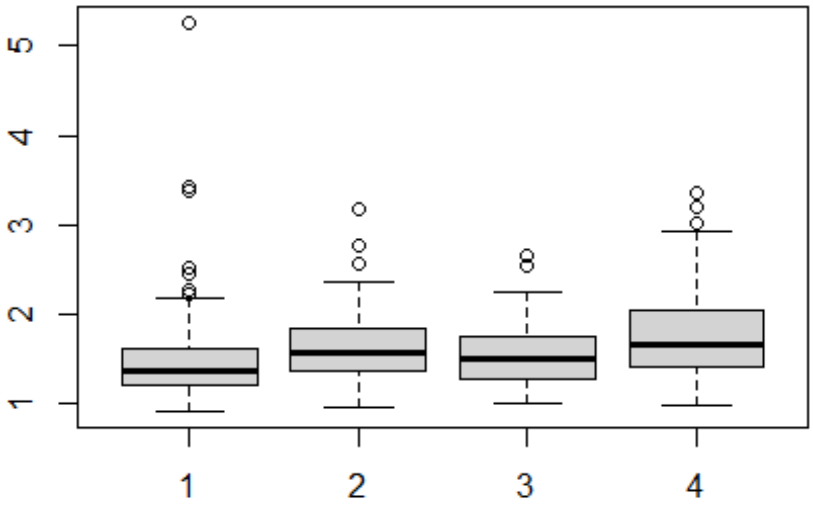

```
boxplot(dat.MCC$auc, dat.MCI$auc, dat.MIC$auc, dat.MII$auc)
```




```
lmt <- as.vector( rbind(mean(dat.MCC$mt),mean(dat.MCI$mt),mean(dat.MIC$mt),
mean(dat.MII$mt) )[,1] )
barplot(lmt, ylim = c(0,2.5), col=c("pink","lightblue"), xaxt="n",ylab =
"Mean MT",legend.text = c("congruent","incongruent"),
        args.legend = list(x = "topright"))
axis(1, at = c(1.4, 0, 3.5, 0), labels = c("80% Congruent","","20%
Congruent", "") ,tick=FALSE, cex=0.5)
```



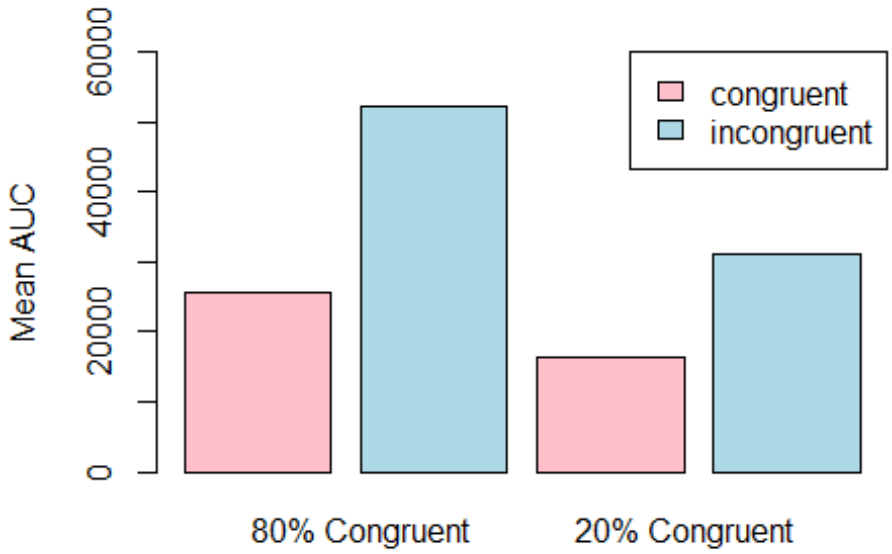

```r
boxplot(dat.MCC$mt, dat.MCI$mt, dat.MIC$mt, dat.MII$mt)
```



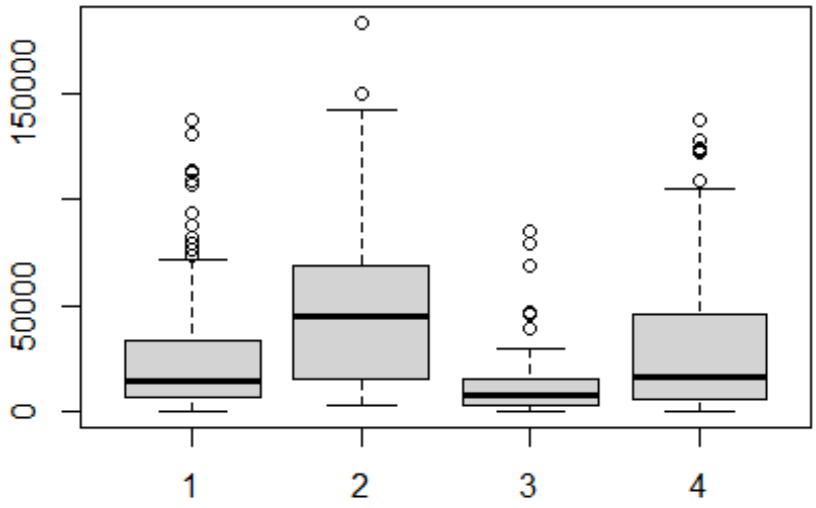

```r
#############################################################3
```

```r
## AUC et MT pour groupe 2


freq <- gp <- auc <- cong <- mt <-it <- c()

for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "C" && dataframe$frequency[i] == "MC" &&
dataframe$groupe[i] == "2"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MCC2 <- data.frame (gp, cong, freq, auc, mt, it)
dat.MCC2 <- na.omit(dat.MCC2)

freq <- gp <- auc <- cong <- mt <-it <- c()
for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "I" && dataframe$frequency[i] == "MC" &&
dataframe$groupe[i] == "2"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MCI2 <- data.frame (gp, cong, freq, auc, mt, it)
dat.MCI2 <- na.omit(dat.MCI2)

freq <- gp <- auc <- cong <- mt <-it <- c()
for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "C" && dataframe$frequency[i] == "MI" &&
dataframe$groupe[i] == "2"){

    cong[i] <- dataframe$congruency[i]
```

```
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MIC2 <- data.frame (gp, cong, freq, auc, mt, it)
dat.MIC2 <- na.omit(dat.MIC2)

freq <- gp <- auc <- cong <- mt <-it <- c()
for (i in 1:nrow(dataframe)) {

  if(dataframe$congruency[i] == "I" && dataframe$frequency[i] == "MI" &&
dataframe$groupe[i] == "2"){

    cong[i] <- dataframe$congruency[i]
    gp[i] <- dataframe$groupe[i]
    auc[i] <- dataframe$AUC[i]
    freq[i] <- dataframe$frequency[i]
    mt[i] <- dataframe$MT[i]
    it[i] <- dataframe$IT[i]

  }
}

dat.MII2 <- data.frame (gp, cong, freq, auc, mt, it)
dat.MII2 <- na.omit(dat.MII2)

lauc2 <- as.vector( rbind(
mean(dat.MCC2$auc),mean(dat.MCI2$auc),mean(dat.MIC2$auc), mean(dat.MII2$auc)
)[,1] )

barplot(lauc2, ylim = c(0,60000), col=c("pink","lightblue"), xaxt="n",ylab =
"Mean AUC",legend.text = c("congruent","incongruent"),
        args.legend = list(x = "topright"))
axis(1, at = c(1.4, 0, 3.6, 0), labels = c("80% Congruent","","20%
Congruent", "") ,tick=FALSE, cex=0.5)
```

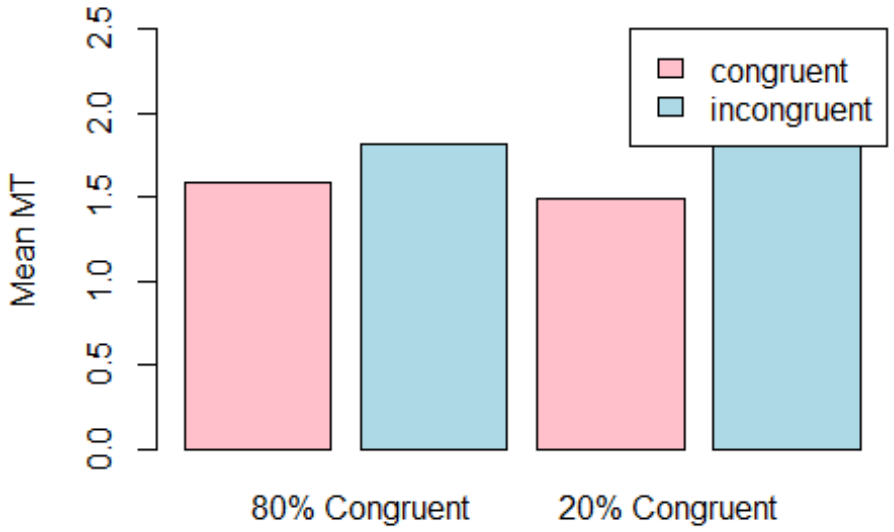



```
boxplot(dat.MCC2$auc,dat.MCI2$auc,dat.MIC2$auc, dat.MII2$auc)
```

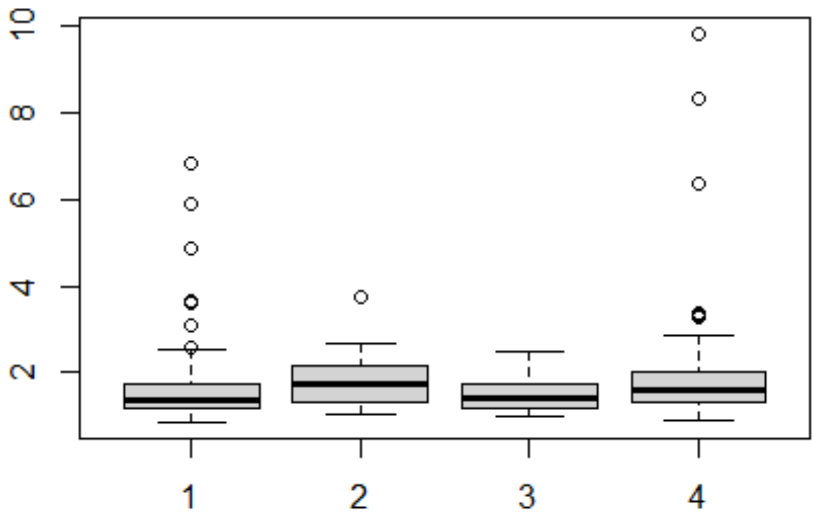



```
lmt2 <- as.vector(
rbind(mean(dat.MCC2$mt),mean(dat.MCI2$mt),mean(dat.MIC2$mt),
```

```
mean(dat.MII2$mt) )[,1] )
barplot(lmt2, ylim = c(0,2.5), col=c("pink","lightblue"), xaxt="n",ylab =
"Mean MT",legend.text = c("congruent","incongruent"),
        args.legend = list(x = "topright"))
axis(1, at = c(1.4, 0, 3.5, 0), labels = c("80% Congruent","","20%
Congruent", "") ,tick=FALSE, cex=0.5)
```




```
boxplot(dat.MCC2$mt, dat.MCI2$mt, dat.MIC2$mt, dat.MII2$mt)
```




```
#################################################################
###anova


# Compute the analysis of variance
mtc.aov <- aov(MT ~ congruency, data = dataframe)
mtf.aov <- aov(MT ~ frequency, data = dataframe)

aucc.aov <- aov(AUC ~ congruency, data = dataframe)
aucf.aov <- aov(AUC ~ frequency, data = dataframe)

itc.aov <- aov(IT ~ congruency, data = dataframe)
itf.aov <- aov(IT ~ frequency, data = dataframe)

# Summary of the analysis
summary(mtc.aov)

##              Df Sum Sq Mean Sq F value   Pr(>F)
## congruency    1   11.5  11.476   22.72 2.27e-06 ***
## Residuals   718  362.7   0.505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(mtf.aov)

##              Df Sum Sq Mean Sq F value Pr(>F)
## frequency     1    5.2   5.160   10.04 0.0016 **
```

```
## Residuals   718  369.0   0.514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aucc.aov)
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## congruency    1 3.366e+10 3.366e+10   31.25 3.23e-08 ***
## Residuals   718 7.736e+11 1.077e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aucf.aov)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## frequency     1 3.745e+09 3.745e+09   3.347 0.0678 .
## Residuals   718 8.035e+11 1.119e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(itc.aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## congruency    1   0.51  0.5097    1.71  0.191
## Residuals   718 214.05  0.2981
```

```
summary(itf.aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## frequency     1   0.67  0.6677   2.241  0.135
## Residuals   718 213.90  0.2979
```

```
#####################################################################3
```

```
## test
```

```
str (part1)
```

```
## 'data.frame':    120 obs. of  10 variables:
##  $ id                 : int  139 139 139 139 139 139 139 139 139 139 ...
##  $ groupe             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ essais.congruent   : int  0 0 1 0 0 1 0 0 0 0 ...
##  $ essais.mot         : chr  "VERT" "JAUNE" "VERT" "VERT" ...
##  $ essais.couleur     : chr  "red" "blue" "green" "red" ...
##  $ essais.reactionTime: num  1.535 1.315 1.375 0.991 1.497 ...
##  $ essais.tempsInit   : num  1.043 0.226 -0.27 0.61 0.361 ...
##  $ essais.courbeX     :List of 120
##   ..$ : int  786 786 786 786 786 786 786 786 786 786 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  756 756 756 756 756 756 756 756 756 756 ...
##   ..$ : int  754 754 754 754 754 754 754 754 754 754 ...
##   ..$ : int  778 778 778 778 778 778 778 778 778 778 ...
```

```
##   ..$ : int  773 773 773 773 773 773 773 773 773 773 ...
##   ..$ : int  771 771 771 771 771 771 771 771 771 771 ...
##   ..$ : int  762 762 762 762 762 762 762 762 762 762 ...
##   ..$ : int  794 794 794 794 794 794 794 794 794 794 ...
##   ..$ : int  790 790 790 790 790 790 790 790 790 790 ...
##   ..$ : int  744 744 744 744 744 744 744 744 744 744 ...
##   ..$ : int  806 806 806 806 806 806 806 806 806 806 ...
##   ..$ : int  781 781 781 781 781 781 781 781 781 781 ...
##   ..$ : int  798 798 798 798 798 798 798 798 798 798 ...
##   ..$ : int  766 766 766 766 766 766 766 766 766 766 ...
##   ..$ : int  781 781 781 781 781 781 781 781 781 781 ...
##   ..$ : int  772 772 772 772 772 772 772 772 772 772 ...
##   ..$ : int  790 790 790 790 790 790 790 790 790 790 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  794 794 794 794 794 794 794 794 794 794 ...
##   ..$ : int  775 775 775 775 775 775 775 775 775 775 ...
##   ..$ : int  794 794 794 794 794 794 794 794 794 794 ...
##   ..$ : int  775 775 774 774 774 774 774 774 774 774 ...
##   ..$ : int  790 790 790 790 790 790 790 790 790 790 ...
##   ..$ : int  974 974 974 974 974 974 974 974 974 974 ...
##   ..$ : int  783 783 783 783 783 783 783 783 783 783 ...
##   ..$ : int  754 754 754 754 754 754 754 754 754 754 ...
##   ..$ : int  786 786 786 786 786 786 786 786 786 786 ...
##   ..$ : int  760 760 760 760 760 760 760 760 760 760 ...
##   ..$ : int  800 800 800 800 800 800 800 800 800 800 ...
##   ..$ : int  794 794 794 794 794 794 794 794 794 794 ...
##   ..$ : int  769 769 769 769 769 769 769 769 769 769 ...
##   ..$ : int  772 772 772 772 772 772 772 772 772 772 ...
##   ..$ : int  774 774 774 774 774 774 774 774 774 774 ...
##   ..$ : int  760 760 760 760 760 760 760 760 760 760 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  779 779 779 779 779 779 779 779 779 779 ...
##   ..$ : int  775 775 775 775 775 775 775 775 775 775 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  741 741 741 742 742 742 742 742 742 742 ...
##   ..$ : int  774 774 774 774 774 774 774 774 774 774 ...
##   ..$ : int  765 765 765 765 765 765 765 765 765 765 ...
##   ..$ : int  775 775 775 775 775 775 775 775 775 775 ...
##   ..$ : int  769 769 769 769 768 768 768 768 768 768 ...
##   ..$ : int  781 781 781 781 781 781 781 781 781 781 ...
##   ..$ : int  755 755 755 755 755 755 755 755 755 755 ...
##   ..$ : int  780 780 780 780 780 780 780 780 780 780 ...
##   ..$ : int  776 776 776 776 776 776 776 776 776 776 ...
##   ..$ : int  801 801 801 801 801 801 801 801 801 801 ...
##   ..$ : int  818 818 818 818 818 818 818 818 818 818 ...
##   ..$ : int  782 782 782 782 782 782 782 782 782 782 ...
##   ..$ : int  785 785 785 785 785 785 785 785 785 785 ...
##   ..$ : int  767 767 767 767 767 767 767 767 767 767 ...
##   ..$ : int  794 794 794 794 794 794 794 794 794 794 ...
##   ..$ : int  748 748 748 748 748 748 748 748 748 748 ...
```

```
##   ..$ : int  779 779 779 779 779 779 779 779 779 779 ...
##   ..$ : int  763 763 763 763 763 763 763 763 763 763 ...
##   ..$ : int  754 754 754 754 754 754 754 754 754 754 ...
##   ..$ : int  793 793 793 793 793 793 793 793 793 793 ...
##   ..$ : int  758 758 758 758 758 758 758 758 758 758 ...
##   ..$ : int  782 782 782 782 782 782 782 782 782 782 ...
##   ..$ : int  768 768 768 768 768 768 768 768 768 768 ...
##   ..$ : int  787 787 787 787 787 787 787 787 787 787 ...
##   ..$ : int  777 777 777 777 777 777 777 777 777 777 ...
##   ..$ : int  798 798 798 798 798 798 798 798 798 798 ...
##   ..$ : int  774 774 774 774 774 774 774 774 774 774 ...
##   ..$ : int  754 754 754 754 754 754 754 754 754 754 ...
##   ..$ : int  786 786 786 786 786 786 786 786 786 786 ...
##   ..$ : int  797 797 797 797 797 797 797 797 797 797 ...
##   ..$ : int  784 784 784 784 784 784 784 784 784 784 ...
##   ..$ : int  750 750 750 750 750 750 750 750 750 750 ...
##   ..$ : int  793 793 793 793 793 793 793 793 793 793 ...
##   ..$ : int  779 779 779 779 779 779 779 779 779 779 ...
##   ..$ : int  786 786 786 786 786 786 786 786 786 786 ...
##   ..$ : int  767 767 767 767 767 767 767 767 767 767 ...
##   ..$ : int  766 766 766 766 766 766 766 766 766 766 ...
##   ..$ : int  763 763 763 763 763 763 763 763 763 763 ...
##   ..$ : int  794 794 794 794 794 794 794 794 794 794 ...
##   ..$ : int  790 790 790 790 790 790 790 790 790 790 ...
##   ..$ : int  754 754 754 754 754 754 754 754 754 754 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  786 786 786 786 786 786 786 786 786 786 ...
##   ..$ : int  774 774 774 774 774 774 774 774 774 774 ...
##   ..$ : int  736 736 736 736 736 736 736 736 736 736 ...
##   ..$ : int  775 775 775 775 775 775 775 775 775 775 ...
##   ..$ : int  759 759 759 759 759 759 759 759 759 759 ...
##   ..$ : int  780 780 780 780 780 780 780 780 780 780 ...
##   ..$ : int  818 818 818 818 818 818 818 818 818 818 ...
##   ..$ : int  780 780 780 780 780 780 780 780 780 780 ...
##   ..$ : int  766 766 766 766 766 766 766 766 766 766 ...
##   ..$ : int  780 780 780 780 780 780 780 780 780 780 ...
##   ..$ : int  770 770 770 770 770 770 770 770 770 770 ...
##   ..$ : int  771 771 771 771 771 771 771 771 771 771 ...
##   ..$ : int  795 795 795 795 795 795 795 795 795 795 ...
##   ..$ : int  772 772 772 772 772 772 772 772 772 772 ...
##   ..$ : int  774 774 774 774 774 774 774 774 774 774 ...
##   ..$ : int  808 808 808 808 808 808 808 808 808 808 ...
##   ..$ : int  776 776 776 776 776 776 776 776 776 776 ...
##   .. [list output truncated]
##  $ essais.courbeY     :List of 120
##   ..$ : int  665 665 665 665 665 665 665 665 665 665 ...
##   ..$ : int  637 637 637 637 637 637 637 637 637 637 ...
##   ..$ : int  618 619 619 620 620 620 620 620 620 620 ...
##   ..$ : int  654 654 654 654 654 654 654 654 654 654 ...
```

```
##   ..$ : int  649 649 649 649 649 649 649 649 649 649 ...
##   ..$ : int  647 647 647 647 647 647 647 647 647 647 ...
##   ..$ : int  639 639 639 639 639 639 639 639 639 639 ...
##   ..$ : int  637 637 637 637 637 637 637 637 637 637 ...
##   ..$ : int  657 657 657 656 656 656 656 656 656 656 ...
##   ..$ : int  639 639 639 639 639 639 639 639 639 639 ...
##   ..$ : int  628 628 628 628 628 628 628 628 628 628 ...
##   ..$ : int  635 635 635 635 635 635 635 635 635 635 ...
##   ..$ : int  632 632 632 632 632 632 632 632 632 632 ...
##   ..$ : int  642 642 642 642 642 642 642 642 642 642 ...
##   ..$ : int  627 627 627 627 627 627 627 627 627 627 ...
##   ..$ : int  645 645 645 645 645 645 645 645 645 645 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  649 649 649 649 649 649 649 649 649 649 ...
##   ..$ : int  650 650 650 650 650 650 650 650 650 650 ...
##   ..$ : int  635 635 635 635 635 635 635 635 635 635 ...
##   ..$ : int  639 639 639 639 639 639 639 639 639 639 ...
##   ..$ : int  641 641 641 641 641 641 641 641 641 641 ...
##   ..$ : int  627 627 627 627 628 628 628 628 629 629 ...
##   ..$ : int  638 638 638 638 638 638 638 638 638 638 ...
##   ..$ : int  158 158 158 158 158 158 158 158 158 158 ...
##   ..$ : int  677 677 677 677 677 677 677 677 677 677 ...
##   ..$ : int  646 646 646 646 646 646 646 646 646 646 ...
##   ..$ : int  662 662 662 662 662 662 662 662 662 662 ...
##   ..$ : int  635 635 635 635 635 635 635 635 635 635 ...
##   ..$ : int  664 664 664 664 664 664 664 664 664 664 ...
##   ..$ : int  647 647 647 647 647 647 647 647 647 647 ...
##   ..$ : int  643 643 643 643 643 643 643 643 643 643 ...
##   ..$ : int  653 653 653 653 653 653 653 653 653 653 ...
##   ..$ : int  635 635 635 635 635 635 635 635 635 635 ...
##   ..$ : int  648 648 648 648 649 649 649 649 649 649 ...
##   ..$ : int  636 636 636 636 636 636 636 636 636 636 ...
##   ..$ : int  622 622 622 622 622 622 622 622 622 622 ...
##   ..$ : int  630 630 630 630 630 630 630 630 630 630 ...
##   ..$ : int  630 630 630 630 630 630 630 630 630 630 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  626 626 627 627 627 627 627 627 627 627 ...
##   ..$ : int  638 638 638 638 638 638 638 638 638 638 ...
##   ..$ : int  620 620 621 621 621 621 621 622 622 622 ...
##   ..$ : int  633 633 633 633 633 633 633 633 633 633 ...
##   ..$ : int  645 645 645 645 646 646 646 646 646 646 ...
##   ..$ : int  642 642 642 642 642 642 642 642 642 642 ...
##   ..$ : int  627 627 627 627 627 627 627 627 627 627 ...
##   ..$ : int  633 633 633 633 633 633 633 633 633 633 ...
##   ..$ : int  645 645 645 645 645 645 645 645 645 645 ...
##   ..$ : int  633 633 633 633 633 633 633 633 633 633 ...
##   ..$ : int  638 638 638 638 638 638 638 638 638 638 ...
##   ..$ : int  638 638 638 638 638 638 638 638 638 638 ...
##   ..$ : int  654 654 654 654 654 654 654 654 654 654 ...
```

```
##   ..$ : int  643 643 643 643 643 643 643 643 643 643 ...
##   ..$ : int  654 654 654 654 654 654 654 654 654 654 ...
##   ..$ : int  643 643 643 643 643 643 643 643 643 643 ...
##   ..$ : int  622 622 622 622 622 622 622 622 622 622 ...
##   ..$ : int  665 665 665 665 665 665 665 665 665 665 ...
##   ..$ : int  629 629 629 629 629 629 629 629 629 629 ...
##   ..$ : int  657 657 657 657 657 657 657 657 657 657 ...
##   ..$ : int  642 642 642 642 642 642 642 642 642 642 ...
##   ..$ : int  645 645 645 645 645 645 645 645 645 645 ...
##   ..$ : int  631 631 631 631 631 631 631 631 631 631 ...
##   ..$ : int  650 650 650 650 650 650 650 650 650 650 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  650 650 650 650 650 650 650 650 650 650 ...
##   ..$ : int  654 654 654 654 654 654 654 654 654 654 ...
##   ..$ : int  641 641 641 641 641 641 641 641 641 641 ...
##   ..$ : int  618 618 618 618 618 618 618 618 618 618 ...
##   ..$ : int  633 633 633 633 633 633 633 633 633 633 ...
##   ..$ : int  641 641 641 641 641 641 641 641 641 641 ...
##   ..$ : int  627 627 627 627 627 627 627 627 627 627 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  626 626 626 626 626 626 626 626 626 626 ...
##   ..$ : int  630 630 630 630 630 630 630 630 630 630 ...
##   ..$ : int  618 618 618 618 618 618 618 618 618 618 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  687 687 687 687 687 687 687 687 687 687 ...
##   ..$ : int  677 677 677 677 677 677 677 677 677 677 ...
##   ..$ : int  641 641 641 641 641 641 641 641 641 641 ...
##   ..$ : int  616 616 616 616 616 616 616 616 616 616 ...
##   ..$ : int  661 661 661 661 661 661 661 661 661 661 ...
##   ..$ : int  628 628 628 628 628 628 628 628 628 628 ...
##   ..$ : int  654 654 654 654 654 654 654 654 654 654 ...
##   ..$ : int  622 622 622 622 622 622 622 622 622 622 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  642 642 642 642 642 642 642 642 642 642 ...
##   ..$ : int  660 660 660 660 660 660 660 660 660 660 ...
##   ..$ : int  675 675 675 675 675 675 675 675 674 674 ...
##   ..$ : int  643 643 643 643 643 643 643 643 643 643 ...
##   ..$ : int  636 636 636 636 636 636 636 636 636 636 ...
##   ..$ : int  628 628 628 628 628 628 628 628 628 628 ...
##   ..$ : int  619 619 619 619 619 619 619 619 619 619 ...
##   ..$ : int  632 632 632 632 632 632 632 632 632 632 ...
##   ..$ : int  634 634 634 634 634 634 634 634 634 634 ...
##   ..$ : int  674 674 674 674 674 674 674 674 674 674 ...
##   ..$ : int  654 654 654 654 654 654 654 654 654 654 ...
##   ..$ : int  625 625 625 625 625 625 625 625 625 625 ...
##   .. [list output truncated]
##  $ essais.erreur      : int  0 0 0 0 0 0 0 0 0 0 ...

names(part1)
```

```
##  [1] "id"                  "groupe"              "essais.congruent"
##  [4] "essais.mot"          "essais.couleur"      "essais.reactionTime"
##  [7] "essais.tempsInit"    "essais.courbeX"      "essais.courbeY"
## [10] "essais.erreur"
```

```
#un tableau qui montre chaque mot et la couleur dans laquelle il existe
table(part1$essais.couleur, part1$essais.mot)
```

```
##
##          BLEU JAUNE ROUGE VERT
##   blue     24     6     0    0
##   green     0     0    24    6
##   red       0     0     5   24
##   yellow    6    25     0    0
```

```
#un tableau qui montre chaque mot et la couleur dans laquelle il existe
table(part2$essais.couleur, part2$essais.mot)
```

```
##
##          BLEU JAUNE ROUGE VERT
##   blue      6    24     0    0
##   green     0     0     6   24
##   red       0     0    24    6
##   yellow   24     6     0    0
```

```
########################################3
```