Algorithmic Analysis for L0TF

1 Load required functions and library

```
library(AMIAS)
library(ggplot2)
library("gridExtra")
```

```
## Warning: package 'gridExtra' was built under R version 3.6.3
```

```
source("utils.R")
```

```
## Warning: package 'ggpubr' was built under R version 3.6.3
```

```
source("amiasutils.R")
```

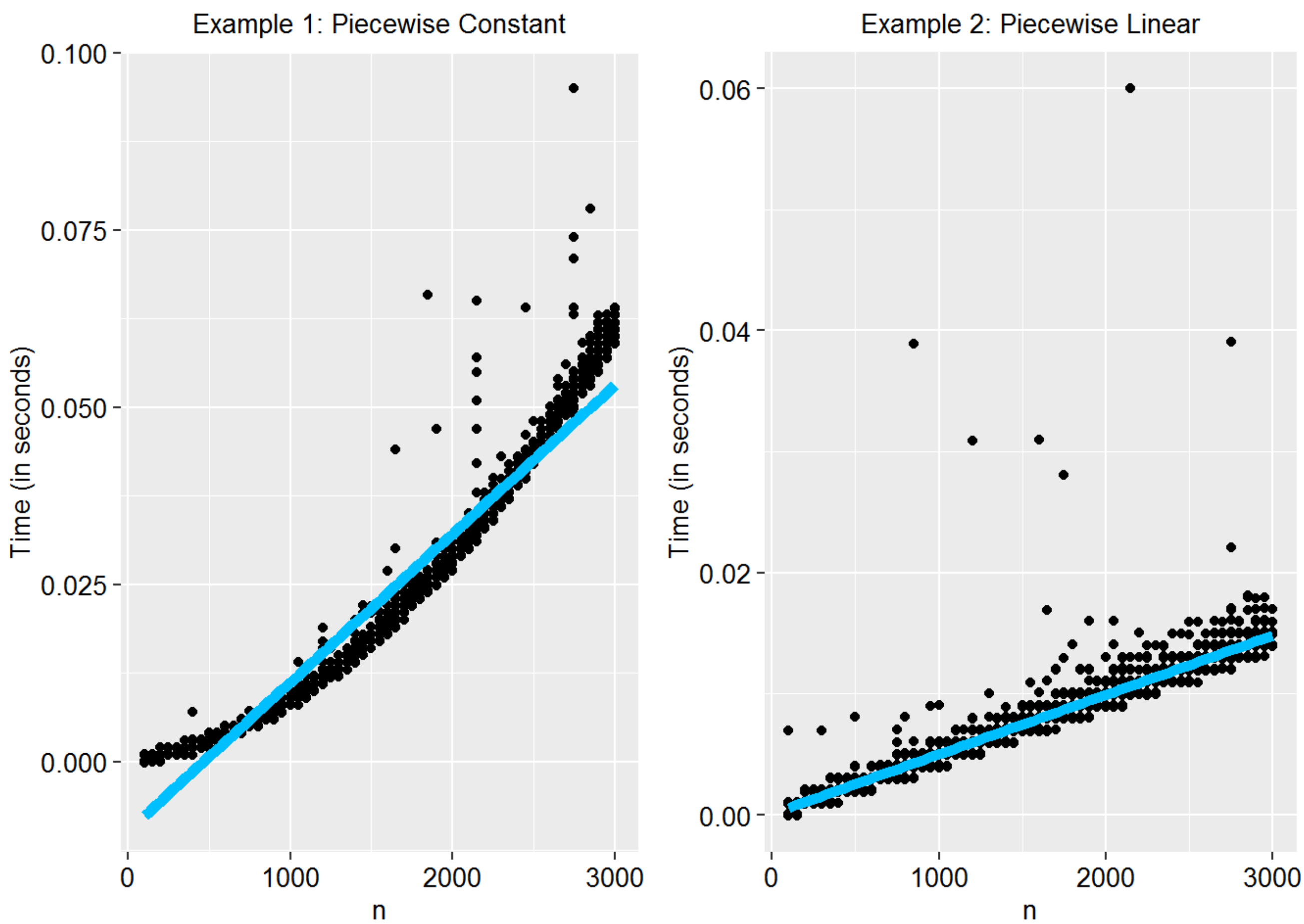
2 Figure 7: Scatterplot of runtime versus sample size

```
nlist <- seq(100, 3000, 50)
runtime <- array(0, c(length( nlist),100, 2))

for(i in seq_along(nlist)){
  n <- nlist[i]
  for(seed in 1:100){

    # Piecewise constant case
    # Toy Piecewise constant/linear: one knot only
    q=0; sigma=0.1; nknot=1
    data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
    start_time <- Sys.time()
    amias(data$y, D_type="tf0", k=nknot)
    end_time <- Sys.time()
    runtime[i, seed, 1] <- end_time - start_time

    q=1; sigma=0.1;  nknot=1
    data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
    start_time <- Sys.time()
    amias(data$y, D_type="tfq", q=q, k=nknot)
    end_time <- Sys.time()
    runtime[i, seed, 2] <- end_time - start_time

  }
}
save.image("runtime_amias.RData")

# load("runtime_amias.RData")

t1 <- as.numeric(t(runtime[,,1]))
data <- cbind(rep(nlist, each=100), t1)
data <- as.data.frame(data)
colnames(data) <- c("n", "time")
library(ggplot2)
p1 <- ggplot(data = data, aes( x= n, y= time)) + geom_jitter(height = 0.0001, width=0) +
  geom_smooth(method=lm, se= FALSE, size=2, color = "deepskyblue") +
  ggtitle("Example 1: Piecewise Constant") +
  labs(x="n", y = "Time (in seconds)") +
  theme(axis.text.x = element_text(size = 10,color="black"),
        axis.title.x = element_text(size = 10,color="black"),
        axis.text.y = element_text(size = 10,color="black"),
        axis.title.y = element_text(size = 10,color="black"),
        plot.title = element_text(size = 10, hjust = 0.5))



t1 <- as.numeric(t(runtime[,,2]))
t1[which.max(t1)] <- NA
```

```
data <- cbind(rep(nlist, each=100), t1)
data <- as.data.frame(data)
colnames(data) <- c("n", "time")
library(ggplot2)
p2 <- ggplot(data = data, aes( x= n, y= time)) + geom_jitter(height = 0.0001, width=0) +
  geom_smooth(method=lm, se= FALSE, size=2, color = "deepskyblue") +
  ggtitle("Example 2: Piecewise Linear") +
  labs(x="n", y = "Time (in seconds)") +
  theme(axis.text.x = element_text(size = 10,color="black"),
        axis.title.x = element_text(size = 10,color="black"),
        axis.text.y = element_text(size = 10,color="black"),
        axis.title.y = element_text(size = 10,color="black"),
        plot.title = element_text(size = 10, hjust = 0.5))



# png("figs/amias_time_n.png", pointsize=6, width=850, height=400, res=120)
grid.arrange(p1, p2, ncol=2)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



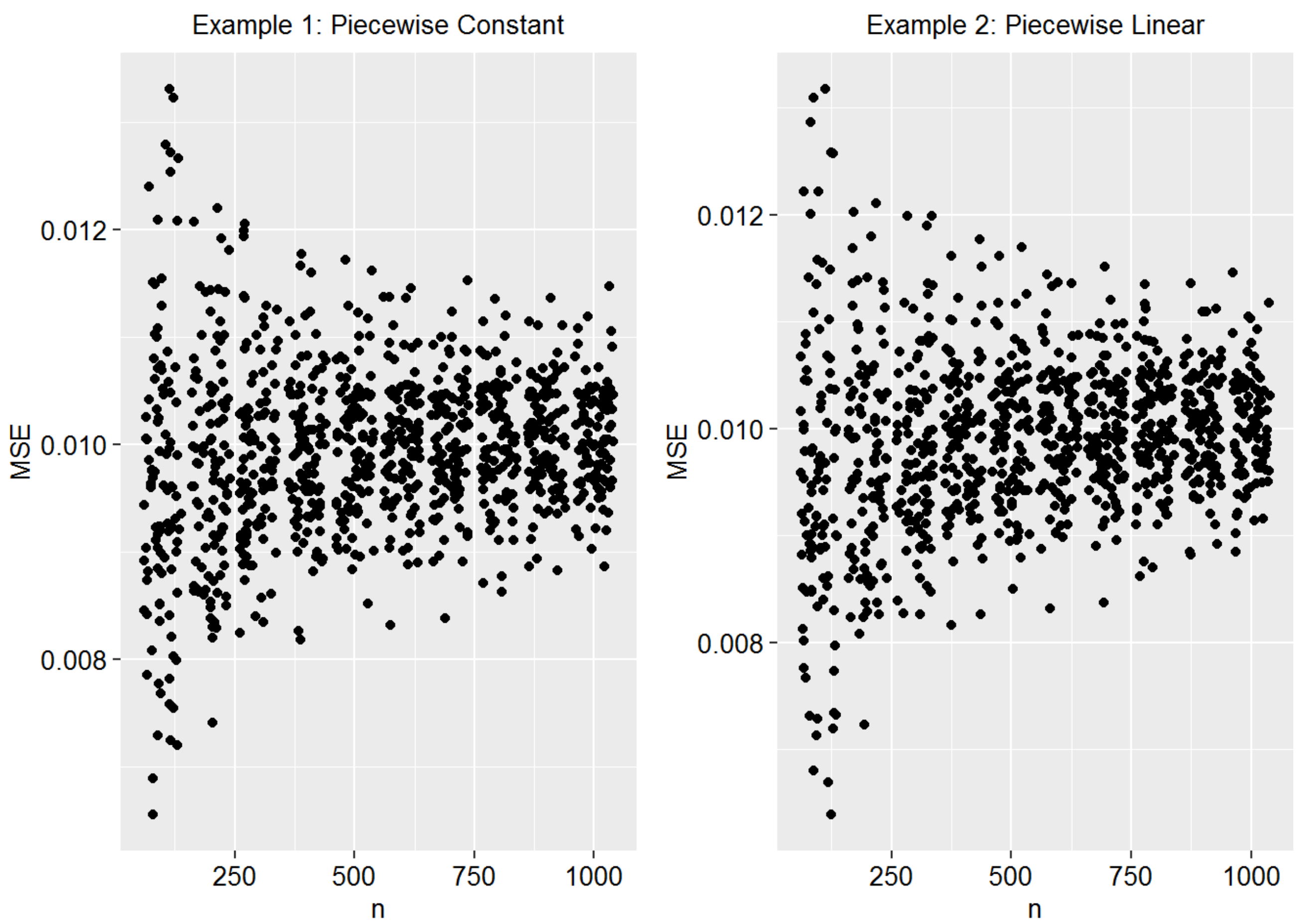
```
# dev.off()
```

3 Figure 8: gitterplot of MSE versus sample size in Example 1 and Example 2

```
nlist <- seq(100, 1000, 100)
mse <- array(0, c(length(nlist),100, 2))

for(i in seq_along(nlist)){
  n <- nlist[i]
  print(n)
  for(seed in 1:100){
    # Piecewise constant case
    sigma=0.1; q=0; nknot = 1
    data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
    resL0 <- amias(data$y, D_type="tf0", k=nknot)
    mse[i,seed,1] <- mean((as.numeric(data$y)-resL0$alpha)^2)

    # Piecewise linear case
    sigma=0.1; q=1; nknot = 1
    data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
    resL0 <- amias(data$y, D_type="tfq", q=q, k=nknot)
    mse[i,seed,2] <- mean((as.numeric(data$y)-resL0$alpha)^2)

  }
}
```

```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000
```

```r
save.image("amias_mse_n_single.RData")

load("amias_mse_n_single.RData")
title <- c("Example 1: Piecewise Constant", "Example 2: Piecewise Linear")
p <- list()
for(j in 1:2){
  t1 <- as.numeric(t(mse[,,j]))
  data <- cbind(rep(nlist, each=100), t1)
  data <- as.data.frame(data)
  colnames(data) <- c("n", "time")
  library(ggplot2)
  p[[j]] <- ggplot(data = data, aes( x= n, y= time)) + geom_jitter()+
    ggtitle(title[j]) +
    labs(x="n", y = "MSE") +
    theme(axis.text.x = element_text(size = 10,color="black"),
          axis.title.x = element_text(size = 10,color="black"),
          axis.text.y = element_text(size = 10,color="black"),
          axis.title.y = element_text(size = 10,color="black"),
          plot.title = element_text(size = 10, hjust = 0.5))
}

# png("figs/amias_mse_n_1.png", pointsize=6, width=850, height=400, res=120)
grid.arrange(p[[1]], p[[2]], ncol=2)
```



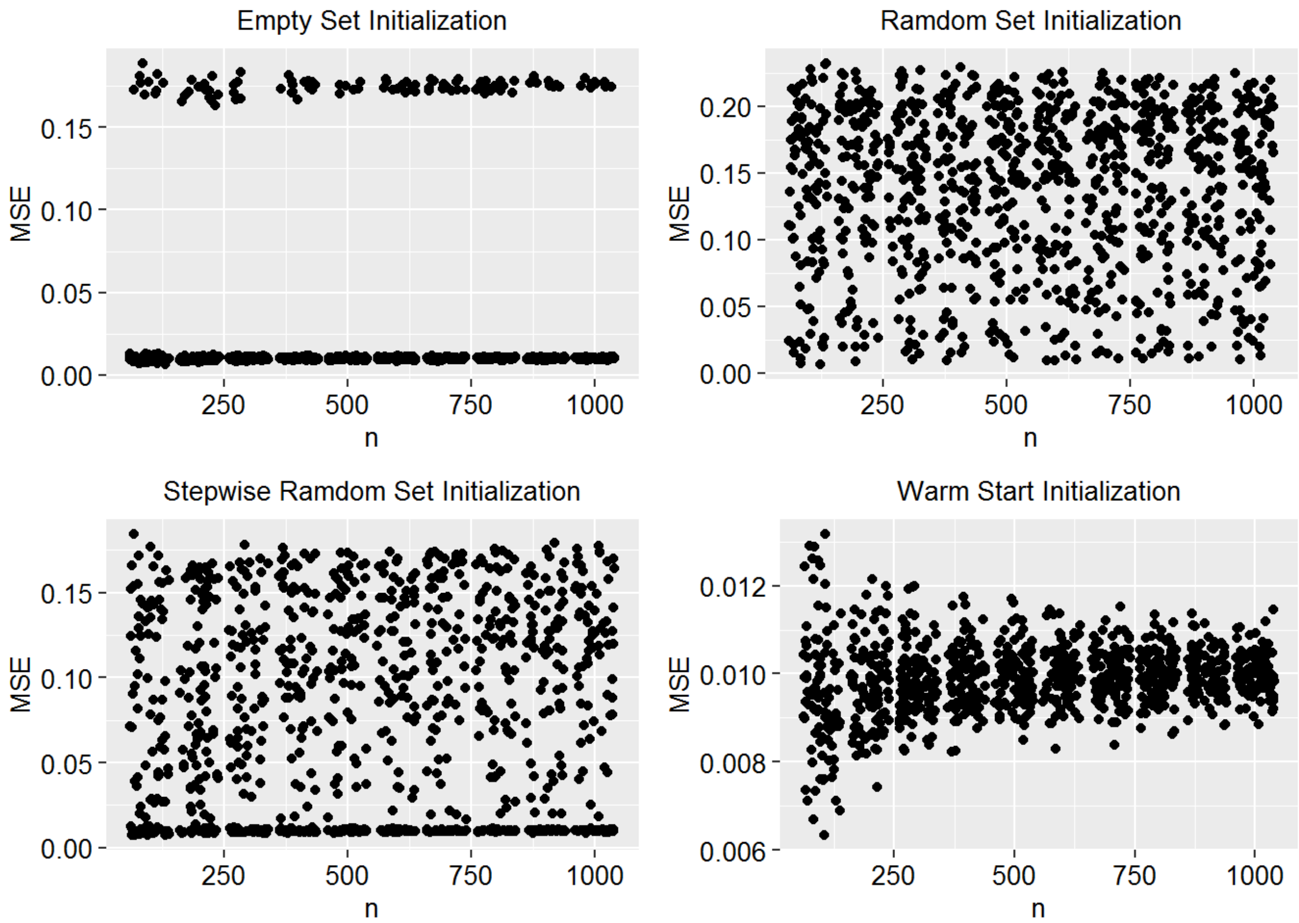
```
# dev.off()
```

4 Figure 9: Gitterplot of MSE versus sample size

```
nlist <- seq(100, 1000, 100)
mse <- array(0, c(length(nlist),100, 4))

for(i in seq_along(nlist)){
  n <- nlist[i]
  print(n)
  for(seed in 1:100){
    # Piecewise constant case
    sigma=0.1; q=0; nknot=2;
    data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)

    # Empty set initialization
    resL0 <- amias(data$y, D_type="tf0", k=nknot)
    mse[i,seed,1] <- mean((as.numeric(data$y)-resL0$alpha)^2)

    # Random set initialization
    resL0 <- amias(data$y, D_type="tf0", k=nknot, A = sample(n-q-1, nknot))
    mse[i,seed,2] <- mean((as.numeric(data$y)-resL0$alpha)^2)

    # Stepwise random set initialization
    resL0_1 <- amias(data$y, D_type="tf0", k=1)
    resL0 <- amias(data$y, D_type="tf0", k=nknot, A = c(resL0_1$A, sample(setdiff(1:(n-q-1), resL0_1$A), 1
)))
    mse[i,seed,3] <- mean((as.numeric(data$y)-resL0$alpha)^2)

    # Warm start initialization
    resL0_1 <- amias(data$y, D_type="tf0", k=1)
    resL0 <- amias(data$y, D_type="tf0", k=nknot, A = c(resL0_1$A, which.max(abs(resL0_1$u))))
    mse[i,seed,4] <- mean((as.numeric(data$y)-resL0$alpha)^2)

  }
}
```

```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000
```

```
save.image("amias_mse_n.RData")

load("amias_mse_n.RData")
title <- c("Empty Set Initialization", "Ramdom Set Initialization", "Stepwise Ramdom Set Initialization", "Warm Start Initialization")
p <- list()
for(j in 1:4){
  t1 <- as.numeric(t(mse[,,j]))
  data <- cbind(rep(nlist, each=100), t1)
  data <- as.data.frame(data)
  colnames(data) <- c("n", "time")
  library(ggplot2)
  p[[j]] <- ggplot(data = data, aes( x= n, y= time)) + geom_jitter()+
    ggtitle(title[j]) +
    labs(x="n", y = "MSE") +
    theme(axis.text.x = element_text(size = 10,color="black"),
          axis.title.x = element_text(size = 10,color="black"),
          axis.text.y = element_text(size = 10,color="black"),
          axis.title.y = element_text(size = 10,color="black"),
          plot.title = element_text(size = 10, hjust = 0.5))
}

# png("figs/amias_mse_n.png", pointsize=6, width=850, height=800, res=120)
grid.arrange(p[[1]], p[[2]],p[[3]], p[[4]], ncol=2)
```



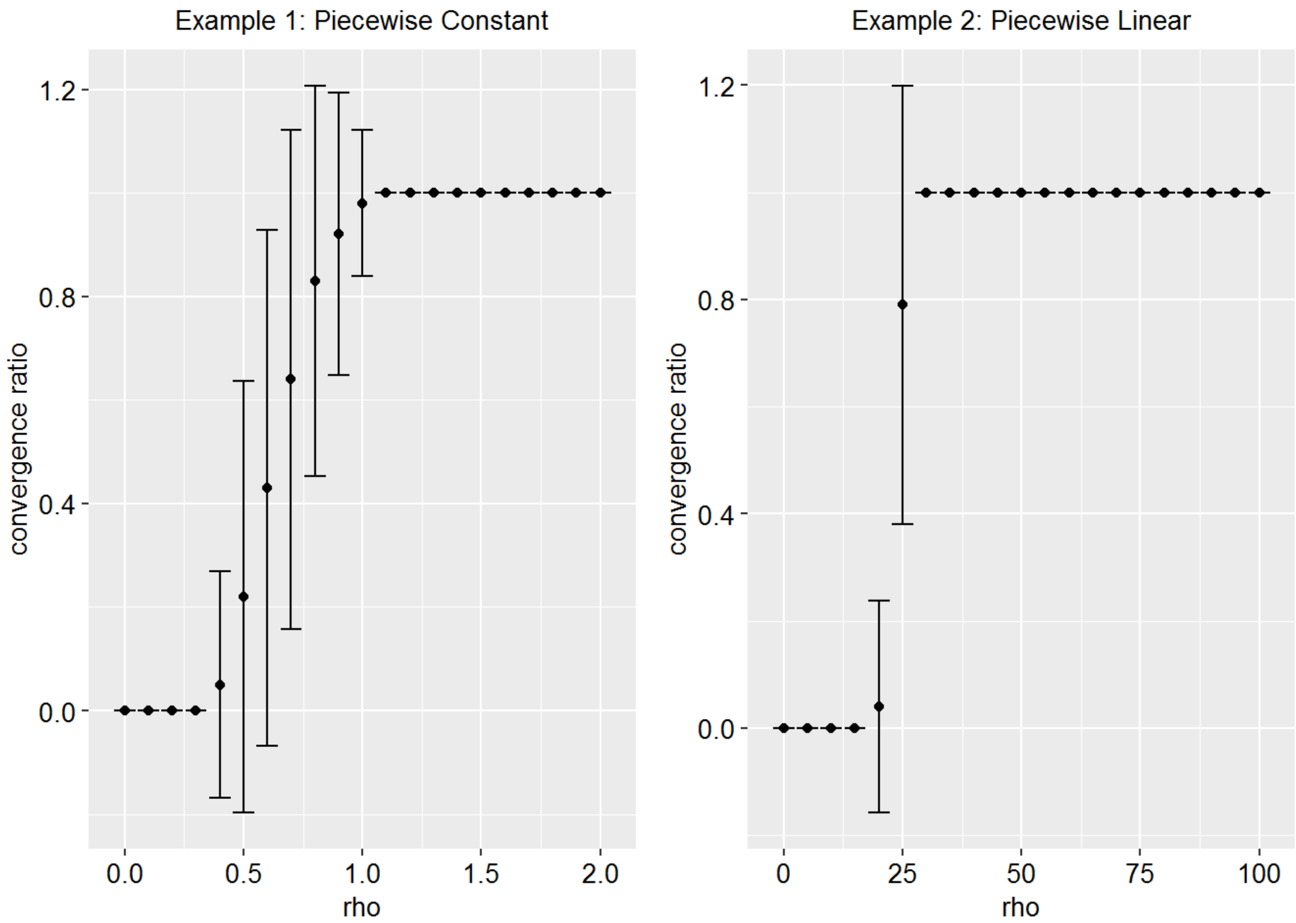

```
# dev.off()
```

5 Figure 10: Error bars of the proportion of convergence against the mixing parameter ρ

```
# amias rho vs convergence
c1 <- c2 <- c()
for(i in 1:100){
  if (i%%20==0) print(i)
  data1 = SimuEx(n=100, sigma=0.1, q=0, nknot=1, seed=i)
  data2 = SimuEx(n=100, sigma=0.1, q=1, nknot=1, seed=i)
  rho1 <- seq(0,2,0.1)
  rho2 <- seq(0,data2$n,5)

  r1 <- r2 <- c()
  for(rho in rho1){
    res1 = amias_R(y = data1$y, D = DiffMat(data1$n, data1$nknot), A = c(), k = data1$nknot, rho = rho, q = 
 data1$q)
    if(res1$iter==20){
      r1 <- c(r1, 0)
    }else{
      r1 <- c(r1, 1)
    }
  }
  for(rho in rho2){
    res2 = amias_R(y = data2$y, D = DiffMat(data2$n, data2$nknot), A = c(), k = data2$nknot, rho = rho, q = 
 data2$q)
    if(res2$iter==20){
      r2 <- c(r2, 0)
    }else{
      r2 <- c(r2, 1)
    }
  }
  c1 <- rbind(c1, r1)
  c2 <- rbind(c2, r2)
}
```

```
## [1] 20
## [1] 40
## [1] 60
## [1] 80
## [1] 100
```

```
p1 = convsteps(t(c1), rho1, "Example 1: Piecewise Constant", 'rho', 'convergence ratio')
p2 = convsteps(t(c2), rho2, "Example 2: Piecewise Linear", 'rho', 'convergence ratio')
save.image("amias_rho_conv.RData")

# png("figs/amias_rho_conv.png", pointsize=8, width=850, height=400, res=120)
ggarrange(p1, p2, ncol = 2)
```



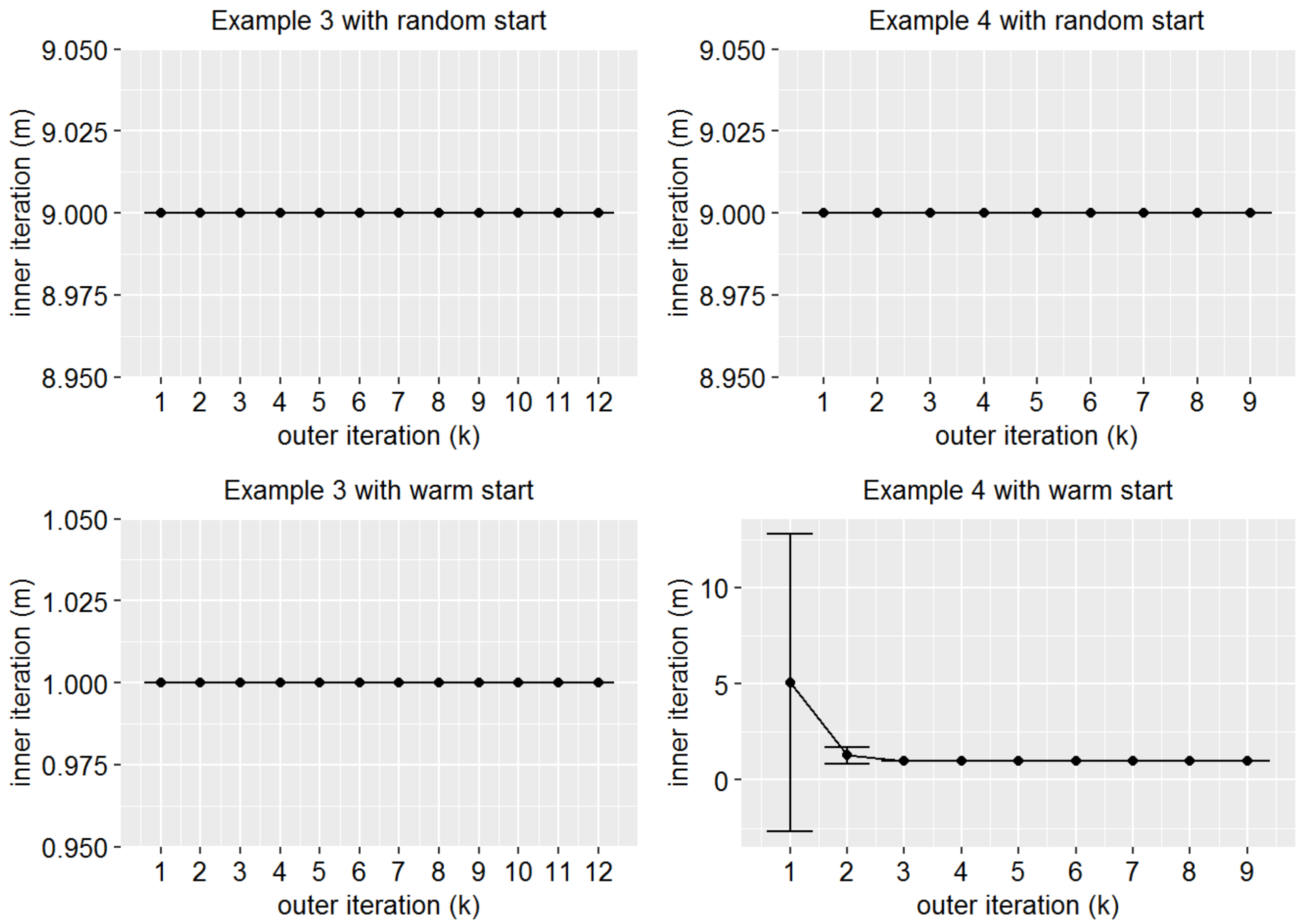

```
# dev.off()
```

6 Figure 11: Error bars of inner iterations against the outer iterations in the sequential AMIAs algorithm

```
library(ggpubr)
c3 <- c4 <- c5 <- c6 <- c()
for(i in 1:100){
  if (i%%20==0) print(i)
  data3 = SimuEx(n=300, sigma=0.1, q=0, nknot=8, seed=i)
  data4 = SimuEx(n=300, sigma=0.1, q=1, nknot=5, seed=i)
  res1 = ramias(data3, kmax = data3$nknot+4)
  res2 = ramias(data4, kmax = data4$nknot+4)
  res3 = samias_R(y = data3$y, D = DiffMat(data3$n, data3$q+1), kmax = data3$nknot+4, rho = data3$n**(data3$q+1), q = data3$q)
  res4 = samias_R(y = data4$y, D = DiffMat(data4$n, data4$q+1), kmax = data4$nknot+4, rho = data4$n**(data4$q+1), q = data4$q)
  c3 = rbind(c3, res1$iters)
  c4 = rbind(c4, res2$iters)
  c5 = rbind(c5, res3$iters)
  c6 = rbind(c6, res4$iters)
}
```

```
## [1] 20
## [1] 40
## [1] 60
## [1] 80
## [1] 100
```

```
p1 = inout_loop(c3, "Example 3 with random start", 'outer iteration (k)', 'inner iteration (m)')
p2 = inout_loop(c4, "Example 4 with random start", 'outer iteration (k)', 'inner iteration (m)')
p3 = inout_loop(c5, "Example 3 with warm start", 'outer iteration (k)', 'inner iteration (m)')
p4 = inout_loop(c6, "Example 4 with warm start", 'outer iteration (k)', 'inner iteration (m)')

save.image("samias_conv_inout.RData")

#  png("figs/samias_conv_inout.png", pointsize=8, width=850, height=850, res=120)
ggarrange(p1, p2, p3, p4, nrow=2, ncol = 2)
```



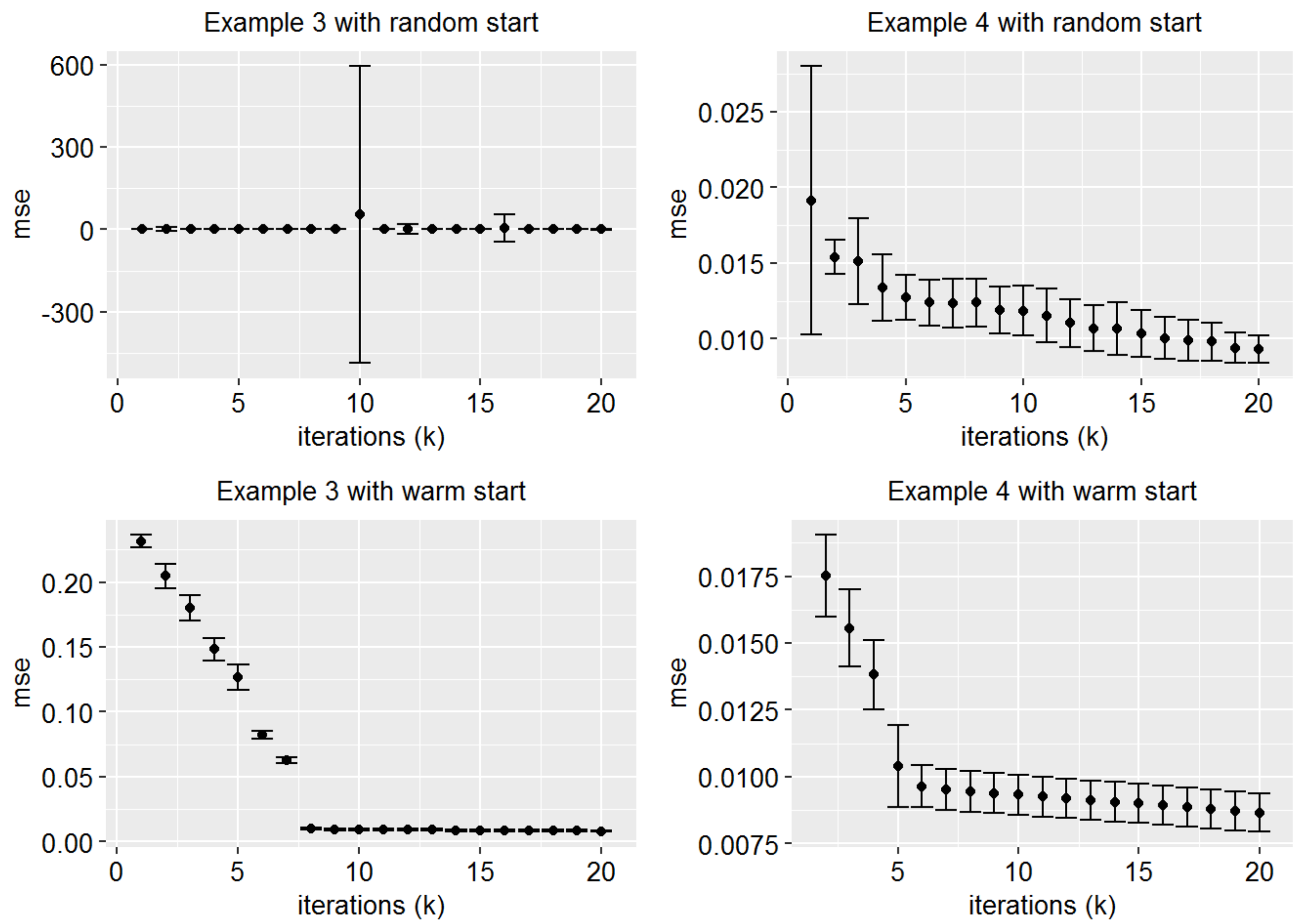

```
# dev.off()
```

7 Figure 12: Error bars of MSE over cardinality parameters

```
c3 <- c4 <- c5 <- c6 <- c()
for(i in 1:100){
  if (i%%20==0) print(i)
  data3 = SimuEx(n=300, sigma=0.1, q=0, nknot=8, seed=i)
  data4 = SimuEx(n=300, sigma=0.1, q=1, nknot=5, seed=i)
  res1 = ramias(data3, kmax = data3$nknot+12)
  res2 = ramias(data4, kmax = data4$nknot+15)
  res3 = samias_R(y = data3$y, D = DiffMat(data3$n, data3$q+1), kmax = data3$nknot+12, rho = data3$n**(data3$q+1), q = data3$q)
  res4 = samias_R(y = data4$y, D = DiffMat(data4$n, data4$q+1), kmax = data4$nknot+15, rho = data4$n**(data4$q+1), q = data4$q)
  c3 = rbind(c3, res1$mse)
  c4 = rbind(c4, res2$mse)
  c5 = rbind(c5, res3$mse)
  c6 = rbind(c6, res4$mse)
}
```

```
## [1] 20
## [1] 40
## [1] 60
## [1] 80
## [1] 100
```

```
p1 = convsteps(t(c3), 1:(data3$nknot+12), "Example 3 with random start", "iterations (k)", "mse")
p2 = convsteps(t(c4), 1:(data4$nknot+15), "Example 4 with random start", "iterations (k)", "mse")
p3 = convsteps(t(c5), 1:(data3$nknot+12), "Example 3 with warm start", "iterations (k)", "mse")
p4 = convsteps(t(c6[,-1]), 2:(data4$nknot+15), "Example 4 with warm start", "iterations (k)", "mse") # cut fi
rst col

save.image("samias_mse_conv.RData")
# png("figs/samias_mse_conv.png", pointsize=8, width=850, height=850, res=120)
ggarrange(p1, p2, p3, p4, nrow=2, ncol = 2)
```



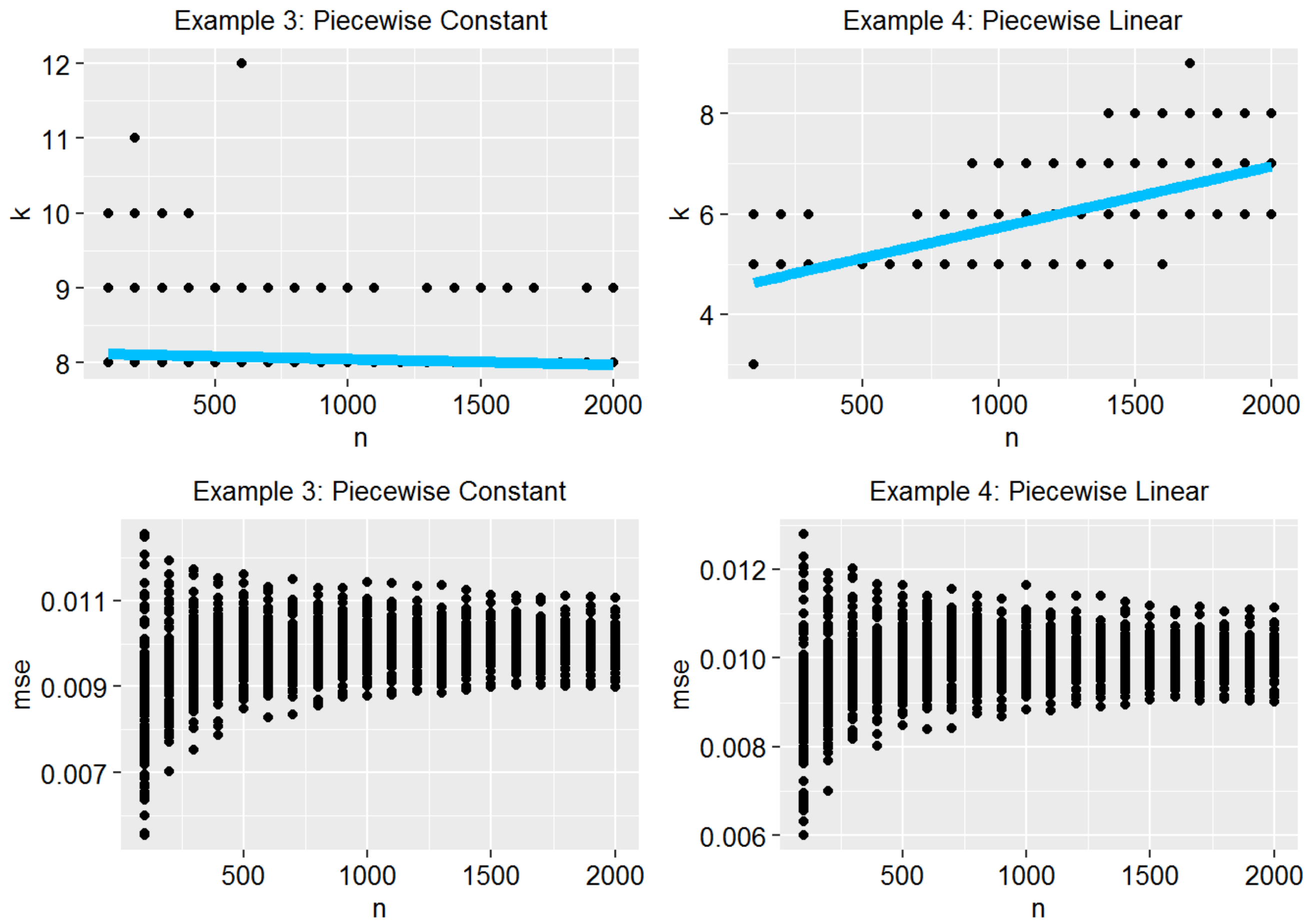

```
# dev.off()
```

8 Figure 13: Scatterplot of number of the dtected knots against sample size

```
nlist <- seq(100, 2000, 100)
re <- re_o <- re_u <- kest <- mse <- array(0, c(length(nlist),100, 2))

for(n in nlist){
  print(n)
  for(seed in 1:100){

    par(mfrow=c(1,2),mar=c(3,3,3,3))
    # Piecewise constant case
    sigma=0.1; q=0; nknot=8;
    data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)
    resL0 = samias(as.numeric(data$y), D_type="tf0", kmax=nknot+4)
    #
    mse[n/100,seed,1] <- mean((as.numeric(data$y)-resL0$alpha)^2)
    kest[n/100,seed,1] <- resL0$kopt
    re[n/100,seed,1] <- nknot==resL0$kopt
    re_o[n/100,seed,1] <- nknot<resL0$kopt
    re_u[n/100,seed,1] <- nknot>resL0$kopt

    sigma=0.1; q=1; nknot=5;
    data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)
    resL0 = samias(as.numeric(data$y), D_type="tfq", q=q, kmax=nknot+4, adjust = TRUE )
    mse[n/100,seed,2] <- mean((as.numeric(data$y)-resL0$alpha)^2)

    kest[n/100,seed,2] <- resL0$kopt

    re[n/100,seed,2] <- nknot==resL0$kopt
    re_o[n/100,seed,2] <- nknot<resL0$kopt
    re_u[n/100,seed,2] <- nknot>resL0$kopt
  }
}
```

```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000
## [1] 1100
## [1] 1200
## [1] 1300
## [1] 1400
## [1] 1500
## [1] 1600
## [1] 1700
## [1] 1800
## [1] 1900
## [1] 2000
```

```
save.image("smias_k_n.RData")

load("smias_k_n.RData")
library(ggplot2)
title <- c("Example 3: Piecewise Constant", "Example 4: Piecewise Linear")
p <- list()
for(j in 1:2){
t1 <- as.numeric(t(kest[,,j]))
data <- cbind(rep(nlist, each=100), t1)
data <- as.data.frame(data)
colnames(data) <- c("n", "k")

p[[j]] <- ggplot(data = data, aes( x= n, y= k)) + geom_point() +
      geom_smooth(method=lm, se= FALSE, size=2, color = "deepskyblue") + ggtitle(title[j]) +
      theme(axis.text.x = element_text(size = 10,color="black"),
            axis.title.x = element_text(size = 10,color="black"),
            axis.text.y = element_text(size = 10,color="black"),
            axis.title.y = element_text(size = 10,color="black"),
            plot.title = element_text(size = 10, hjust = 0.5))
}

p2 <- list()
for(j in 1:2){
  t1 <- as.numeric(t(mse[,,j]))
  data <- cbind(rep(nlist, each=100), t1)
  data <- as.data.frame(data)
  colnames(data) <- c("n", "mse")

  p2[[j]] <- ggplot(data = data, aes( x= n, y= mse)) + geom_point() +
       ggtitle(title[j]) +
      theme(axis.text.x = element_text(size = 10,color="black"),
            axis.title.x = element_text(size = 10,color="black"),
            axis.text.y = element_text(size = 10,color="black"),
            axis.title.y = element_text(size = 10,color="black"),
            plot.title = element_text(size = 10, hjust = 0.5))
}

# png("figs/samias_k_n.png", pointsize=8, width=850, height=850, res=120)
grid.arrange(p[[1]], p[[2]], p2[[1]], p2[[2]], ncol=2)
```



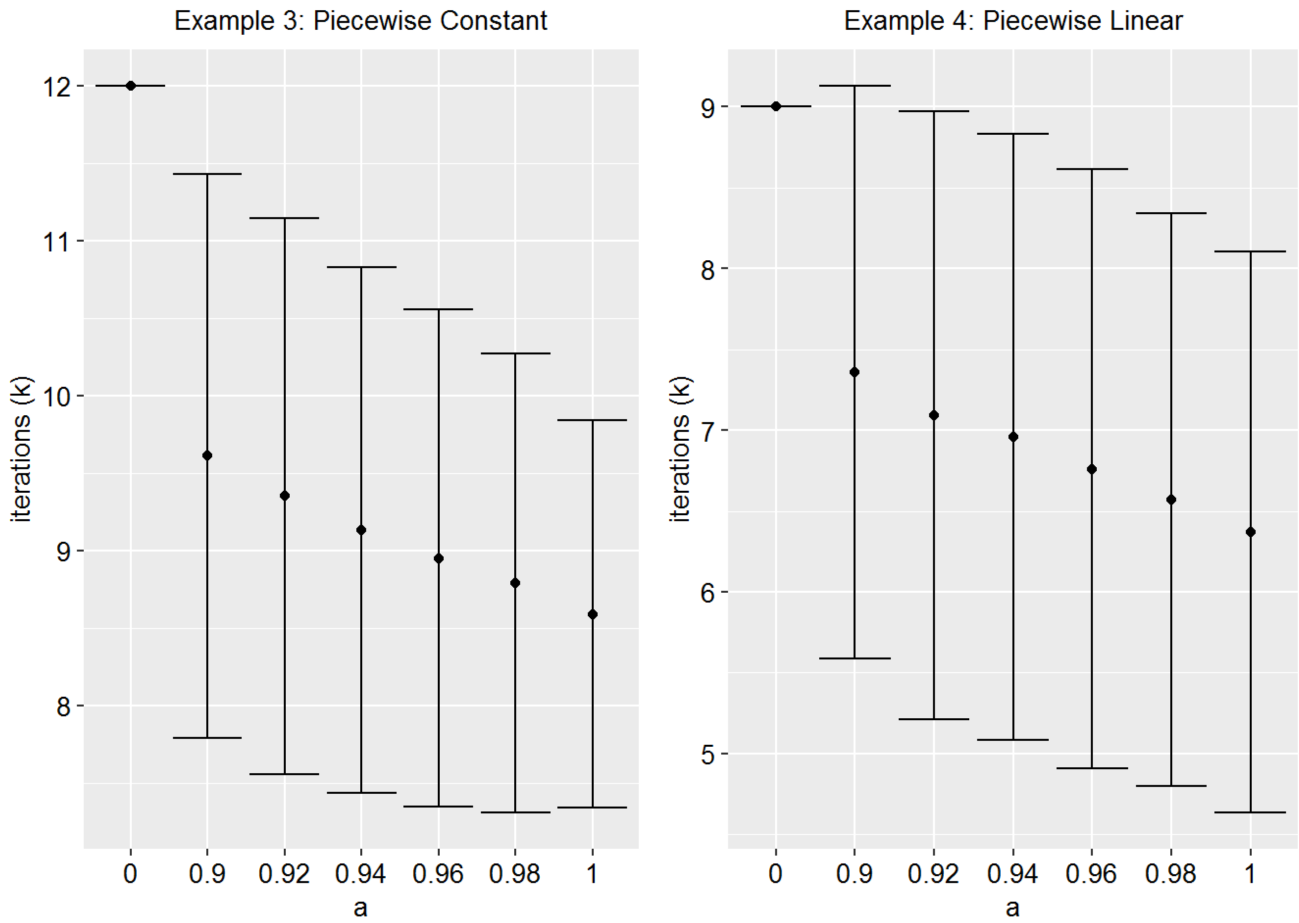

```
# dev.off()
```

9 Figure 14: Early stoppint rule effect on the convergence of sequential AMIAS algorithm

```
# samias eps vs convergence
c1 = c2 = c()
for(i in 1:100){
  if(i%%20==0) print(i)
  data3 = SimuEx(n=300, sigma=0.1, q=0, nknot=8, seed=i)
  data4 = SimuEx(n=300, sigma=0.1, q=1, nknot=5, seed=i)
  r1 <- r2 <- c()
  sig1 <- median(abs(diff(data3$y,diff=1)))/(qnorm(3/4)*sqrt(choose(2,1)))
  sig2 <- median(abs(diff(data4$y,diff=2)))/(qnorm(3/4)*sqrt(choose(4,2)))

  eps1 <- c(0,seq(0.9,1,0.02))*(sig1^2)
  eps2 <- c(0,seq(0.9,1,0.02))*(sig2^2)
  for(eps in eps1){
    res1 = samias_R(y = data3$y, D = DiffMat(data3$n, data3$q+1), kmax = data3$nknot+4, rho = data3$n**(data
3$q+1), q = data3$q, eps=eps)
    r1 <- c(r1, res1$outiter)
  }
  for(eps in eps2){
    res2 = samias_R(y = data4$y, D = DiffMat(data4$n, data4$q+1), kmax = data4$nknot+4, rho = data4$n**(data
4$q+1), q = data4$q, eps=eps)
    r2 <- c(r2, res2$outiter)
  }
  c1 <- rbind(c1, r1)
  c2 <- rbind(c2, r2)
}
```

```
## [1] 20
## [1] 40
## [1] 60
## [1] 80
## [1] 100
```

```
eps1 <- c(0,seq(0.9,1,0.02))
p1 = convsteps(t(c1), as.character(eps1), "Example 3: Piecewise Constant", "a", "iterations (k)")
p2 = convsteps(t(c2), as.character(eps1), "Example 4: Piecewise Linear", "a", "iterations (k)")
save.image("samias_eps_conv.RData")

load("samias_eps_conv.RData")
# png("figs/samias_eps_conv.png", pointsize=8, width=850, height=400, res=120)
ggarrange(p1, p2, ncol = 2)
```



Example 3: Piecewise Constant
iterations (k)
12
11
10
9
8
0
0.9
0.92
0.94
0.96
0.98
1
a
Example 4: Piecewise Linear
iterations (k)
9
8
7
6
5
0
0.9
0.92
0.94
0.96
0.98
1
a


```
# dev.off()
```