

## Speeding up computation: parallel computing and Rcpp

Biao Zhang, December 14<sup>th</sup>, 2021

Q1.

R codes:

```
setwd('H:\\MATH796\\week 11\\')
library(Rcpp)
test_1 = c(1:5, 5:1, 1:5)
sourceCpp('10_Lab_Rcpp.cpp')
which_max_cpp(test_1)
which.max(test_1)
```

Output:

```
> which_max_cpp(test_1)
[1] 5 6 15
> which.max(test_1)
[1] 5
```

My codes return all indexes that the max is reached.

Q2.

R codes:

```
sourceCpp('10_Lab_Rcpp.cpp')
eps = 1e-12
iter.max = 1000
iter=0
err.x=1
x.init = matrix(c(5,1),ncol = 1)
x.all = x.init
while(err.x>eps&iter<iter.max){
  x.new = update_newton_cpp(x.init,.8)
  err.x = sqrt(sum((x.new-x.init)^2))
  x.all = cbind(x.all,x.new)
  iter=iter+1
  x.init=x.new
}

if(iter==iter.max) print(paste0('The algorithm does not converge in ',iter.max,' iterations!'))
if(err.x<eps) print(paste0('The algorithm converges at x= (',paste0(x.new,collapse = ','),') at iteration ',iter,''))
```

Output:

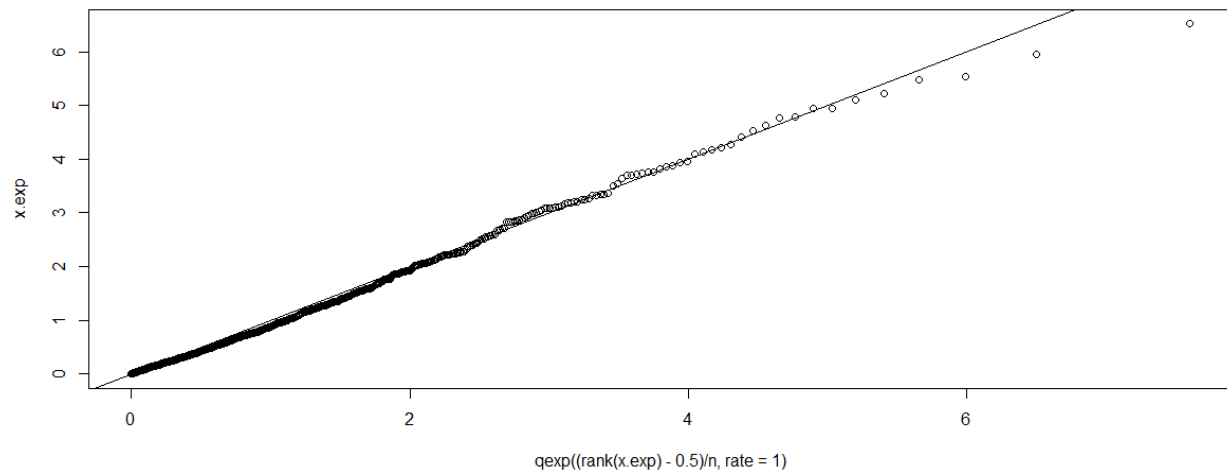
```
> if(err.x<eps) print(paste0('The algorithm converges at x= (',paste0(x.new,collapse = ','),') at iteration ',iter,''))
[1] "The algorithm converges at x= (0,0) at iteration 2."
```

Q3.

R codes:

```
sourceCpp('10_Lab_Rcpp.cpp')
set.seed(100)
n = 1000
x.exp = ar_sampling_exp_cpp(n)
plot(qexp((rank(x.exp)-0.5)/n,rate = 1),x.exp)
abline(a=0,b=1)
```

Output:



I used `sourceCpp()` for all my functions. Their codes could be found in the attached [10\\_Lab\\_Rcpp.cpp](#) file.