Hw3

Kevin Luo

2019/9/22

```r
A1 <- diag(c(1,2,2), nrow = 3)
A2 <- diag(c(1,2,0), nrow = 3)
b <- matrix(c(1,1,0), ncol = 1)
epsilon <- c(1e-8,1e-8,1e-8)
lambda <- 0.1
converge <- function(X, lastX){
  for (i in 1:length(X)){
    if (abs(X[i]-lastX[i])>epsilon){
      return(FALSE)
    }
  }
  return(TRUE)
}
for (i in 1:5){
  X = rnorm(3)
  lastX <- X+1
  while (converge(X, lastX) == FALSE){
    lastX = X
    X = X - lambda*(A1 %*% X - b)
  }
  print(X)
}
```

```
##               [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -3.013299e-16
##               [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -3.698993e-15
##               [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -1.410955e-15
##               [,1]
## [1,]  1.000000e+00
## [2,]  5.000000e-01
## [3,] -5.005869e-14
##               [,1]
## [1,]  9.999999e-01
## [2,]  5.000000e-01
## [3,] -2.037503e-17
```

```r
# Compare the results to $x* = A1^{-1}b$
solve(A1) %*%b
```

```
##      [,1]
## [1,]  1.0
## [2,]  0.5
## [3,]  0.0
```

```r
# They converged to the same x*
```

But that is not the case when A is not invertible. Mathematically, the optimization problem has infinite set of solutions. The converged result will thus depends on the initialization process.

```r
for (i in 1:5){
  X = rnorm(3)
  lastX <- X+1
  while (converge(X, lastX) == FALSE){
    lastX = X
    X = X - lambda*(A2 %*% X - b)
  }
  print(X)
}
```

```
##           [,1]
## [1,] 0.9999999
## [2,] 0.5000000
## [3,] 1.1574709
##            [,1]
## [1,]  0.9999999
## [2,]  0.5000000
## [3,] -0.8228044
##            [,1]
## [1,]  0.9999999
## [2,]  0.5000000
## [3,] -0.3215081
##            [,1]
## [1,]  0.9999999
## [2,]  0.5000000
## [3,] -1.2682850
##           [,1]
## [1,] 0.9999999
## [2,] 0.5000000
## [3,] 1.5508908
```