# Ejercicios

Análisis Numérico

Joaquin Cavieres G.

## Factorización LU

Si A es una matriz de dimensión $n \times n$ y sí A puede ser factorizada como $A = LU$ con $L$ una matriz triangular inferior y todas las diagonales iguales a 1, y $U$ una matriz triangular superior, entonces $LU$ se dice que es una descomposición $LU$ de $A$. Cabe señalar que puede que no sea posible obtener una descomposición $LU$ de $A$, pero se puede mostrar existe una matriz de permutación $P$ tal que $A = PLU$, o equivalentemente, $P^T A = LU$ ya que todas las matrices de permutación son ortogonales. El efecto de premultiplicar $A$ con la matriz $P^T$ es reordenar las filas de $A$, lo que nos permite decir que la factorización es única.

*Observación:* Algunos autores (por ejemplo, Banerjee y Roy, 2014) restringen la definición de una descomposición $LU$ a matrices no singulares, en donde todas las entradas diagonales de $U$ son distintas de cero.

### Ejemplo 1

Una manera sencilla de hacer la descomposición $LU$ es mediante la librería `Matrix`, la cual calcula la descomposición de $A$ usando la matriz de permutación $P$.

```
library(Matrix)
A = matrix(c(1,2,3,4,5,6,5,4,3,2,1,2,-3,2,5,20), ncol = 4)
det(A)                 # -144 => A es no singular!!
```

```
## [1] -144
```

```
mat_lu = lu(A)
expand(mat_lu)         # Muestra a P, L, U
```

```
## $L
## 4 x 4 Matrix of class "dtrMatrix" (unitriangular)
##      [,1]       [,2]       [,3]       [,4]
## [1,]  1.0000000          .          .          .
## [2,]  0.5000000  1.0000000          .          .
## [3,]  0.2500000  1.0000000  1.0000000          .
## [4,]  0.7500000  0.5000000 -0.6666667  1.0000000
##
## $U
```

```
## 4 x 4 Matrix of class "dtrMatrix"
##      [,1] [,2] [,3] [,4]
## [1,]  4.0  4.0  2.0 20.0
## [2,]    .  4.0  1.0 -8.0
## [3,]    .    .  1.5  0.0
## [4,]    .    .    . -6.0
##
## $P
## 4 x 4 sparse Matrix of class "pMatrix"
##
## [1,] . . | .
## [2,] . | . .
## [3,] . . . |
## [4,] | . . .
```

```r
A2 = matrix(1, nrow = 3,ncol = 3) # ejemplo de una matriz no invertible
lu(A2)
```

```
## 'MatrixFactorization' of Formal class 'denseLU' [package "Matrix"] with 4 slots
##   ..@ x       : num [1:9] 1 1 1 1 0 0 1 0 0
##   ..@ perm    : int [1:3] 1 2 3
##   ..@ Dimnames:List of 2
##   .. ..$ : NULL
##   .. ..$ : NULL
##   ..@ Dim     : int [1:2] 3 3
```

Ahora, para comparar resultados, generamos una función para el la factorización $LU$:

```r
LU <- function(A){
  r <- nrow(A)
  L <- diag(r)#create an identity matrix of the same size as A
  U <- A
  i <- 2
  j <- 1
  k <- 1
  while(k <= r) {
    while (i <= r) {
      if (U[i,j]!=0) {
        multiplier <- (-U[i,j]/U[k,j])
        L[i,j] <- -multiplier
            while (j <= r) {
              U[i,j] <- U[i,j] + U[k,j]*multiplier
              j <- j+1
            }
        }
          i <- i+1
          j <- 1
```

```
    }
    k <- k+1
    i <- k+1
    j <- k
  }
  matrixList <- list(L,U)
  return (matrixList)
}
```

Utilizamos la función `LU` creada con una matriz de $3 \times 3$

```
A <- matrix(c(2,1,-6,4,-4,-9,-4,3,5),nrow = 3)
L <- LU(A)[[1]]
U <- LU(A)[[2]]
```

Vemos los resultados

```
L
```

```
##      [,1] [,2] [,3]
## [1,]  1.0  0.0    0
## [2,]  0.5  1.0    0
## [3,] -3.0 -0.5    1
```

```
U
```

```
##      [,1] [,2] [,3]
## [1,]    2    4 -4.0
## [2,]    0   -6  5.0
## [3,]    0    0 -4.5
```

```
L%*%U
```

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

```
A
```

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