

Segunda parte

Carlos Souza
2023-12-13

Eliminação de Gauss

Questão 1

```
a <- matrix(c(2,1,3,5,-2, -3,2,-4,2,3, 4,-1,1,-2,2, 5,3,2,4,-1, -6,4,-1,3,5), nrow = 5, ncol = 5)
n <- nrow(a)
b <- c(-21,29,-2,30,24)
```

Item 1-a)

```
cat("Matriz Inicial A|b:\n")
```

Matriz Inicial A|b:

```
print(cbind(a, b))
```

b
[1,] 2 -3 4 5 -6 -21
[2,] 1 2 -1 3 4 29
[3,] 3 -4 1 2 -1 -2
[4,] 5 2 -2 4 3 30
[5,] -2 3 2 -1 5 24

```
for (k in 1:(n - 1)) {
  cat("\nA(",k,") e b(",k,"):\n")

  for (i in (k + 1):n) {
    m = a[i, k] / a[k, k]
    a[i, k] = 0

    for (j in (k + 1):n) {
      a[i, j] = a[i, j] - m * a[k, j]
    }
    b[i] = b[i] - m * b[k]
  }

  print(cbind(a, b))
}
```

##
A(1) e b(1) :
b
[1,] 2 -3.0 4 5.0 -6 -21.0
[2,] 0 3.5 -3 0.5 7 39.5
[3,] 0 0.5 -5 -5.5 8 29.5
[4,] 0 9.5 -12 -8.5 18 82.5
[5,] 0 0.0 6 4.0 -1 3.0
##
A(2) e b(2) :
b
[1,] 2 -3.0 4.000000 5.000000 -6 -21.00000
[2,] 0 3.5 -3.000000 0.500000 7 39.50000
[3,] 0 0.0 -4.571429 -5.571429 7 23.85714
[4,] 0 0.0 -3.857143 -9.857143 -1 -24.71429
[5,] 0 0.0 6.000000 4.000000 -1 3.00000
##
A(3) e b(3) :
b
[1,] 2 -3.0 4.000000 5.000000 -6.00000 -21.00000
[2,] 0 3.5 -3.000000 0.500000 7.00000 39.50000
[3,] 0 0.0 -4.571429 -5.571429 7.00000 23.85714
[4,] 0 0.0 0.000000 -5.156250 -6.90625 -44.84375
[5,] 0 0.0 0.000000 -3.312500 8.18750 34.31250
##
A(4) e b(4) :
b
[1,] 2 -3.0 4.000000 5.000000 -6.00000 -21.00000
[2,] 0 3.5 -3.000000 0.500000 7.00000 39.50000
[3,] 0 0.0 -4.571429 -5.571429 7.00000 23.85714
[4,] 0 0.0 0.000000 -5.156250 -6.90625 -44.84375
[5,] 0 0.0 0.000000 0.000000 12.62424 63.12121

Item 1-b)

```
# Obs: 1
# Estou considerando duas casas decimais para calcular a solução da matriz
```

```
a <- matrix(c(2,0,0,0,0, -3,3.5,0,0,0, 4,-3,-4.57,0,0, 5,0.5,-5.57,-5.16,0, -6,7,7,-6.91,12.62), nrow = 5, ncol = 5)
cat("Matriz Triangular superior: \n")
```

Matriz Triangular superior:

```
print(a)
```

[,1] [,2] [,3] [,4] [,5]
[1,] 2 -3.0 4.00 5.00 -6.00
[2,] 0 3.5 -3.00 0.50 7.00
[3,] 0 0.0 -4.57 -5.57 7.00
[4,] 0 0.0 0.00 -5.16 -6.91
[5,] 0 0.0 0.00 0.00 12.62

```
b <- c(-21,39.5,23.86,-44.84,63.12) # Ajuste aqui
cat("\n Vetor b: \n")
```

Vetor b:

```
print(b)
```

[1] -21.00 39.50 23.86 -44.84 63.12

```
n <- nrow(a)
x <- as.numeric()
x[n] <- b[n] / a[n, n]
for (k in (n - 1):1) {
  s <- 0
  for (j in (k + 1):n) {
    s <- s + a[k, j] * x[j]
  }

  x[k] <- (b[k] - s) / a[k, k]
}
cat("\nA solucao da matriz e:", "X[1] =", x[1], "X[2] =", x[2], "X[3] =", x[3], "X[4] =", x[4], "X[5] =", x[5],
"\n")
```

##
A solucao da matriz é: X[1] = 1.012898 X[2] = 1.008336 X[3] = 0.01210027 X[4] = 1.992064 X[5] = 5.001585

Fatoração de Cholesky

Questão 3

```
A <- matrix(c(1,2,-1, 2,5,-3, -1,-3,6), nrow = 3, ncol = 3)
B <- c(-5,-13.4,16)
n <- nrow(A)
```

```
G <- matrix(rep(0, n*n), ncol = n, nrow = n) # definindo a matriz G
G[1,1] <- A[1,1]^(1/2)
for(i in 2:n){
  G[i,1] <- A[i,1] / G[1,1]
}

for(k in 2:(n - 1)){
  soma <- 0

  for(j in 1:(k - 1)){
    soma <- soma + G[k,j]^2
  }

  r <- A[k,k] - soma
  G[k,k] <- r^(1/2)

  for(i in (k + 1):n){
    soma <- 0

    for(j in 1:(k - 1)){
      soma <- soma + G[i,j] * G[k,j]
    }

    G[i,k] <- (A[i,k] - soma) / G[k,k]
  }
}

soma <- 0
for(j in 1:(n-1)){
  soma <- soma + G[n,j]^2
}

r <- A[n,n] - soma
G[n,n] <- r^(1/2)
```

Item 3-a)

```
cat("\n", "Valor da Matriz G:\n")
```

##
Valor da Matriz G:

```
print(G)
```

[,1] [,2] [,3]
[1,] 1 0 0
[2,] 2 1 0
[3,] -1 -1 2

```
cat("\n", "Valor da Matriz transposta de G':\n")
```

##
Valor da Matriz transposta de G':

```
print(t(G))
```

[,1] [,2] [,3]
[1,] 1 2 -1
[2,] 0 1 -1
[3,] 0 0 2

Item 3-b)

```
##### Gy = b

a <- matrix(c(1,2,-1, 0,1,-1, 0,0,2), nrow = 3, ncol = 3)
b <- c(-5,-13.4,16)

n <- nrow(a)
x <- numeric()
```

```
x[1] <- b[1]/a[1,1]
for (i in 2:n) {
  s <- 0
  for (j in 1:(i-1)) {
    s <- s + a[i, j] * x[j]
  }
  x[i] <- (b[i] - s) / a[i,i]
}
```

```
cat("A solucao da matriz Gy = b: ", "y[1] =", x[1],",", "y[2] =",x[2],",", "y[3] =", x[3],"\n")
```

A solucao da matriz Gy = b: y[1] = -5 ; y[2] = -3.4 ; y[3] = 3.8

Item 3-c)

```
##### G'x = y

a <- matrix(c(5,1,0,0, -2,0,3,0, 0,2,7,3, -1,0,-2,7), nrow = 3, ncol = 3)
b <- c(-5,-3.4,3.0)

x <- as.numeric()
n <- nrow(a)
x[n] <- b[n]/a[n,n]
```

```
for(k in (n-1):1){
  s <- 0
  for(i in (k+1):n){
    s <- s + a[k,j]*x[j]
  }
  x[k] <- (b[k] - s) / a[k,k]
}
```

```
cat("A solucao da matriz G'x = y: ", "X[1] =",x[1],",", "X[2] =",x[2],",", "X[3] =", x[3], "\n")
```

A solucao da matriz G'x = y: X[1] = -0.1 ; X[2] = -1.5 ; X[3] = 1.9

Questão 4

Item 4-b)

```
##### Gauss-Jacobi

a <- matrix(c(5,1,0,0, -2,0,3,0, 0,2,7,3, -1,0,-2,7), nrow = 4, ncol = 4) # Matriz
b <- c(2,-11,-27,29) # Vetor

precisao <- 0.001
X0 <- c(1,1,1,1) # chute inicial
n <- nrow(a);
```

```
valor_max <- Inf # Variável para guardar em cada iteração o valor maximo
k <- 0 # iterações

C <- matrix(0, ncol = n, nrow = n) # Definindo a matriz C
G <- numeric(n) # Definindo o vetor G

for (i in 1:n) { # Método para interagir com as linhas e colunas
  for (j in 1:n) {
    if (i != j) { # verificar se a posição linha e colunas são diferentes
      C[i, j] <- -a[i, j] / a[i, i]
    }
  }
  G[i] <- b[i] / a[i, i] # Guardar valores na matriz G
}
```

```
while (valor_max > precisao) { # Verificar se o valor maximo é menor que o erro adotado

  X1 <- C %*% X0 + G
  d <- abs(X1 - X0) # Método de Gauss-Jacobi
  valor_max <- max(d)
  X0 <- X1
  k <- k + 1

  cat("Iteração:", k, "\n")
  cat("Vetor X:", X0, "\n")
  cat("Valor máximo do vetor de X:", valor_max, "\n\n")
}
```

Iteração: 1
Vetor X: 1 -1.75 -4 3.714286
Valor máximo do vetor de X: 5
##
Iteração: 2
Vetor X: 0.4428571 -0.5 -2.045918 5.857143
Valor máximo do vetor de X: 2.142857
##
Iteração: 3
Vetor X: 1.371429 -0.9188776 -1.969388 5.019679
Valor máximo do vetor de X: 0.9285714
##
Iteração: 4
Vetor X: 1.036385 -1.054082 -2.029144 4.98688
Valor máximo do vetor de X: 0.3350437
##
Iteração: 5
Vetor X: 0.9757434 -0.9972621 -1.980571 5.01249
Valor máximo do vetor de X: 0.0606414
##
Iteração: 6
Vetor X: 1.003593 -1.001825 -1.997605 4.991673
Valor máximo do vetor de X: 0.02784979
##
Iteração: 7
Vetor X: 0.9976045 -1.001040 -2.001594 4.998973
Valor máximo do vetor de X: 0.007308343
##
Iteração: 8
Vetor X: 0.9993755 -0.9993014 -1.999844 5.000684
Valor máximo do vetor de X: 0.001778992
##
Iteração: 9
Vetor X: 1.000416 -0.9999609 -2.000104 4.999933
Valor máximo do vetor de X: 0.00104083
##
Iteração: 10
Vetor X: 1.0000002 -1.0000026 -2.000036 5.000045
Valor máximo do vetor de X: 0.0004140473

```
# O ultimo valor obtido de cada iteração
cat("O ultimo valor obtido de cada iteração realizada! \n")
```

O ultimo valor obtido de cada iteração realizada!

```
cat("Total de iterações realizadas:", k, "\n")
```

Total de iterações realizadas: 10

```
cat("Valor máximo do vetor de X:", valor_max, "\n\n")
```

Valor máximo do vetor de X: 0.0004140473

```
cat("Vetor X:", X0, "\n")
```

Vetor X: 1.0000002 -1.0000026 -2.000036 5.000045

Item 4-c)

```
##### Gauss-Seidel

a <- matrix(c(5,1,0,0, -2,0,3,0, 0,2,7,3, -1,0,-2,7), nrow = 4, ncol = 4) # Matriz
b <- c(2,-11,-27,29) # Vetor

precisao <- 0.001 # erro
X0 <- c(1,1,1,1) # Chute inicial
n <- nrow(a)
```

```
k <- 0
d <- numeric(n) # guardar os valores

while (valor_max > precisao) {
  for (i in 1:n) {
    soma <- 0 # Método para interagir com as linhas e colunas
    for (j in 1:n) {
      if (j != i) { # verificar se a posição linha e colunas são diferentes
        soma <- soma + a[i, j] * X0[j]
      }
    }
    x1 <- (b[i] - soma) / a[i, i]
    d[i] <- abs(x1 - X0[i])
    X0[i] <- x1
  }
  valor_max <- max(d) # Critério de parada
  k <- k + 1

  cat("Iteração:", k, "\n")
  cat("X:", X0, "\n")
  cat("Valor máximo:", valor_max, "\n\n")
}
```

Iteração: 1
X: 1 -1.75 -2.821429 5.352041
Valor máximo: 4.352041
##
Iteração: 2
X: 0.7704082 -0.7659439 -1.999727 4.999883
Valor máximo: 0.9840561
##
Iteração: 3
X: 1.093599 -1.011768 -1.99499 4.997853
Valor máximo: 0.3231909
##
Iteração: 4
X: 0.9948633 -1.00061 -2.000352 5.000151
Valor máximo: 0.09873574
##
Iteração: 5
X: 0.999786 -0.9998553 -2.000006 5.000003
Valor máximo: 0.004922707
##
Iteração: 6
X: 1.000046 -1.000004 -1.999997 4.999999
Valor máximo: 0.0002604173

```
# O ultimo valor obtido de cada iteração
cat("O ultimo valor obtido de cada iteração realizada! \n")
```

O ultimo valor obtido de cada iteração realizada!

```
cat("Total de iterações realizadas:", k, "\n")
```

Total de iterações realizadas: 6

```
cat("Valor máximo do vetor de X:", valor_max, "\n")
```

Valor máximo do vetor de X: 0.0002604173

```
cat("Vetor X:", X0, "\n")
```

Vetor X: 1.000046 -1.000004 -1.999997 4.999999