# Algoritmos para Solução de Sistemas Lineares

DIM0404 - Cálculo Numérico para Ciência da Computação

Fernanda Menezes Paes Isabel

Abril de 2017

## 1 Introdução

Este trabalho consiste na elaboração de algoritmos para diferentes métodos de resolução de sistemas lineares e suas implementações.

## 2 Algoritmos

### 2.1 Fatorização LU com pivoteamento parcial

Input: Matriz $A$ e vetor $b$.
Ouput: Vetor $x$ que é solução do sistema linear $Ax = b$

---

$n =$ tamanho de $A$
$I =$ matriz identidade de tamanho $n$
**for** $i = 1, ..., n-1$ **do**
    $l =$ linha do valor máximo absoluto da coluna $i$ entre as linhas $i$ e $n$
    **if** $l > i$ **then**
        **for** $j = 1, ..., n$ **do**
            trocar valor de $A_{ij}$ com $A_{lj}$
        **end for**
        trocar valor de $b_i$ com $b_l$
    **end if**
    $G = I$
    **for** $j = i+1, ..., n$ **do**
        $G_{ji} = -\frac{A_{ji}}{A_{ii}}$
    **end for**
    $A = G * A$
    $b = G * b$
**end for**
$x =$ vetor de tamanho $n$
**for** $i = n, ..., 1$ **do**
    $x_i = b_i$
    **for** $j = i+1, ..., n$ **do**
        $x_i = x_i - A_{ij}x_j$
    **end for**
    $x_i = \frac{x_i}{A_{ii}}$
**end for**
**return** $x$

---

## 2.2 Fatoração de Choleski

Input: Matriz simétrica e positiva-definida $A$ e vetor $b$.
Ouput: Vetor $x$ que é solução do sistema linear $Ax = b$

---

$n =$ tamanho de $A$
$I =$ matriz identidade de tamanho $n$
$L^{-1} = I$
**for** $i = 1, ..., n-1$ **do**
  $G = I$
  **for** $j = i+1, ..., n$ **do**
    $G_{ji} = -\frac{A_{ji}}{A_{ii}}$
  **end for**
  $L^{-1} = G * L^{-1}$
  $A = G * A$
  $b = G * b$
**end for**
$D = diag(U)$
**return** $L^{-T} D^{-1} b$

---

## 2.3 Método de Jacobi

Input: Matriz $A$, vetor $b$, vetor inicial $prev$ e $\epsilon > 0$, tal que $||D^{-1}(A - D)|| < 1$, onde $D = diag(A)$.
Ouput: Vetor $x$ que é uma solução aproximada do sistema linear $Ax = b$

---

$n =$ tamanho de $A$
$norma = \epsilon$
**while** $norma >= \epsilon$ **do**
  **for** $i = 1, ..., n$ **do**
    $x_i = -\frac{1}{A_{ii}} \left[ \left( \sum_{j=1}^{n} A_{ij} prev_j \right) - A_{ii} prev_i - b_i \right]$
  **end for**
  $norma = ||x - prev||$
  $prev = x$
**end while**
**return** $x$

---

## 2.4 Método de Gauss-Seidel

Input: Matriz $A$, vetor $b$, vetor inicial $prev$ e $\epsilon > 0$, tal que $||D^{-1}(A - D)|| < 1$, onde $D = diag(A)$.
Ouput: Vetor $x$ que é uma solução aproximada do sistema linear $Ax = b$

$n =$ tamanho de $A$
$norma = \epsilon$
**while** $norma >= \epsilon$ **do**
  **for** $i = 1, ..., n$ **do**
    $x_i = -\frac{1}{A_{ii}}\left[\left(\sum_{j=1}^{i-1} A_{ij}x_j\right) + \left(\sum_{j=i+1}^{n} A_{ij}prev_j\right) - b_i\right]$
  **end for**
  $norma = ||x - prev||$
  $prev = x$
**end while**
**return** $x$

# 3 Comparação dos algoritmos

## 3.1 LU com pivoteamento parcial e Cholesky

Ao testar para um sistema com uma matriz A de dimensão 1000x1000, simétrica e positiva-definida, ou seja, no qual pode ser usado tanto a fatorização LU quanto a Cholesky, podemos observar que a segunda é mais eficiente computacionalmente (roda em menos tempo).

Resultado de testes (tempo em segundos):

```
>> main
tempo LU:
  142.0463

tempo Cholesky:
  140.4002

>> main
tempo LU:
  140.2838

tempo Cholesky:
  121.9098
```

## 3.2 Jacobi e Gauss-Seidel

Ao testar para um sistema com uma matriz A de dimensão 1000x1000, com $||D^{-1}(A - D)|| < 1$, ou seja, no qual pode ser usado tanto o método de Jacobi quanto o de Gauss-Seidel, podemos observar que o segundo é mais eficiente computacionalmente (roda em menos passos e, consequentemente, em menos tempo).

Resultado de testes (tempo em segundos):

Com $\epsilon = 0.1$:

```
>> main
passos Jacobi:
     5

tempo Jacobi:
    0.1953

passos Gauss-Seidel:
     3

tempo Gauss-Seidel:
    0.1140
```

Com $\epsilon = 0.001$:

```
>> main
passos Jacobi:
    12

tempo Jacobi:
    0.4619

passos Gauss-Seidel:
     5

tempo Gauss-Seidel:
    0.2017
```

Com $\epsilon = 0.00001$:

```
>> main
passos Jacobi:
    19

tempo Jacobi:
    0.6300

passos Gauss-Seidel:
     7

tempo Gauss-Seidel:
    0.2498
```

Também podemos ver a importância dos métodos indiretos: apesar do resultado não ser exato, ser uma aproximação, a diferença de tempo para os métodos diretos é gigantesca: de mais de dois minutos nos diretos para menos de um segundo nos indiretos.