# AULA PRÁTICA 2

## ALGORITMOS DE RESOLUÇÃO DE SISTEMAS LINEARES
## ALGORITMOS ITERATIVOS

1) Implementar uma função Scilab resolvendo um sistema linear $Ax = b$ usando o algoritmo iterativo de Jacobi.
   A função deve ter como variáveis de entrada:
   - a matriz A;
   - o vetor b;
   - uma aproximação inicial $x_0$ da solução do sistema;
   - uma tolerância E;
   - um número máximo de iterações M;
   - o tipo de norma a ser utilizada: 1, 2 ou %inf.

   Como variáveis de saída:
   - a solução $x_k$ do sistema encontrada pelo método;
   - a norma da diferença entre as duas últimas aproximações ($||x_k - x_{k-1}||$);
   - o número k de iterações efetuadas;
   - a norma do resíduo ($||r_k|| = ||b - Ax_k||$).

   Critério de parada do algoritmo: use “$||x_k - x_{k-1}|| < E$ ou $k > M$”.

2) Implementar uma função Scilab resolvendo um sistema linear $Ax = b$ usando o algoritmo iterativo de Gauss-Seidel.
   A função deve ter como variáveis de entrada:
   - a matriz A;
   - o vetor b;
   - uma aproximação inicial $x_0$ da solução do sistema;
   - uma tolerância E;
   - um número máximo de iterações M;
   - o tipo de norma a ser utilizada: 1, 2 ou %inf.

   Como variáveis de saída:
   - a solução $x_k$ do sistema encontrada pelo método;
   - a norma da diferença entre as duas últimas aproximações ($||x_k - x_{k-1}||$);
   - o número k de iterações efetuadas;
   - a norma do resíduo ($||r_k|| = ||b - Ax_k||$).

   Critério de parada do algoritmo: use “$||x_k - x_{k-1}|| < E$ ou $k > M$”.

Faça duas implementações diferentes: uma usando a função “inv” do Scilab para calcular a inversa de L+D, obtendo assim a matriz do método $M_G = -(L+D)^{-1}U$ e o vetor $c_G = (L+D)^{-1}b$ para fazer as iterações $x_{k+1} = M_G*x_k + c_G$ e outra resolvendo o sistema linear $(L+D) * x_{k+1} = -U * x_k + b$ para fazer as iterações (a matriz L+D é triangular inferior; escreva uma função para resolver sistemas em que a matriz dos coeficientes é triangular inferior e use-a a cada iteração).

3) Teste as funções implementadas para resolver o sistema

$$\begin{cases} x - 4y + 2z &= 2 \\ \quad 2y + 4z &= 1 \\ 6x - y - 2z &= 1 \end{cases}.$$

Use o vetor $x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ como aproximação inicial.

Agora reordene as equações do sistema dado, de modo que a matriz dos coeficientes seja estritamente diagonal dominante e teste novamente as funções implementadas. Comente os resultados.

4) a) Para o sistema do exercício 3 da Lista de Exercícios 2, mostre que o método de Jacobi com $\mathbf{x}^{(0)}=\mathbf{0}$ falha em dar uma boa aproximação após 25 iterações.

b) Use o método de Gauss-Seidel com $\mathbf{x}^{(0)}=\mathbf{0}$ para obter uma aproximação da solução do sistema linear com precisão de $10^{-5}$ na norma-infinito.

5) a) Utilize o método iterativo de Gauss-Seidel para obter uma aproximação da solução do sistema linear do exercício 5 da Lista de Exercícios 2, com tolerância de $10^{-2}$ e o máximo de 300 iterações.

b) O que acontece ao repetir o item a) quando o sistema é alterado para

$$\begin{cases} x_1 & & -2x_3 & = 0{,}2 \\ -\frac{1}{2}x_1 & +x_2 & -\frac{1}{4}x_3 & = -1{,}425 \\ x_1 & -\frac{1}{2}x_2 & +x_3 & = 2 \end{cases}.$$

6) Agora gere matrizes $A_{nxn}$ com diagonal estritamente dominante para n=10, n=100, n=1000, n=2000, ... bem como vetores b com dimensões compatíveis e resolva esses sistemas Ax=b pelo Método de Gauss-Seidel, usando as duas versões implementadas no item 2. Use as funções tic() e toc() do Scilab para medir os tempos de execução e compará-los.