Lab 9: Métodos Iterativos para Sistemas Lineares

INF1608 – Análise Numérica

Departamento de Informática, PUC-Rio

Para estes exercícios, considere a representação de matrizes quadradas $A_{n\times n}$ como um vetor de vetores do Lab 0.

1. Para a solução de sistemas lineares na forma $A\mathbf{x} = \mathbf{b}$, onde assume-se que A é uma matriz *estritamente diagonal dominante*, pede-se:

   (a) Implemente o método iterativo de Gauss-Seidel:

   $$A = L + D + U$$
   $$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - U\mathbf{x}_k - L\mathbf{x}_{k+1})$$

   Na prática, basta atualizar o vetor solução *in place*, componente a componente:

   $$x_i = \frac{b_i - \sum_{j\neq i} A_{ij}x_j}{A_{ii}}$$

   Escreva uma função que receba como parâmetros a dimensão do problema n, a matriz A, o vetor independente b, a aproximação inicial da solução x e a tolerância de erro aceitável. A iteração deve terminar quando a norma-2 do vetor resíduo ($\mathbf{r} = \mathbf{b} - A\mathbf{x}$) tiver valor menor ou igual à tolerância. A função deve armazenar a solução final em x e retornar o número de iterações efetuado. O protótipo da função é dado por:

   ```
   int gaussseidel (int n, double** A, double* b, double* x, double tol);
   ```

   (b) Seguindo a mesma interface, implemente o método iterativo de Gauss-Seidel com sobre-relaxamento ($w > 1.0$). Isto é, a função deve avaliar cada $\mathbf{x}_{k+1}[i]$ com o método de Gauss-Seidel e ajustar $\mathbf{x}_{k+1}[i]$ com a fórmula do relaxamento:

   $$\mathbf{x}_{k+1}[i] = (1 - w)\,\mathbf{x}_k[i] + w\,\mathbf{x}_{k+1}[i]$$

   A função também deve receber o fator de relaxamento, w, como parâmetro adicional. O protótipo da função é dado por:

   ```
   int sor_gaussseidel (int n, double** A, double* b, double* x,
                        double tol, double w);
   ```

2. Teste, analise e compare a eficiência dos métodos achando as soluções dos sistemas abaixo, usando tolerância 10^{-7}, fator de relaxamento igual a 1.1 e solução inicial igual ao vetor nulo. O programa de teste exibe na tela o número de iterações e a solução encontrada para cada um dos métodos.

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \quad , \quad \begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix} \quad e$$

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0.5 \\ -1 & 3 & -1 & 0 & 0.5 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0.5 & 0 & -1 & 3 & -1 \\ 0.5 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 1.5 \\ 1 \\ 1 \\ 1.5 \\ 2.5 \end{bmatrix}$$

Sabe-se que as soluções destes sistemas são $[1,\ 2]$, $[2,\ -1,\ 1]$ e $[1,\ 1,\ 1,\ 1,\ 1,\ 1]$, respectivamente.

Agrupe os protótipos das funções pedidas em um módulo "metiter.h" e as implementações em um módulo "metiter.c". Escreva o teste em outro módulo "main.c".

Entrega: O código fonte deste trabalho (isto é, os arquivos "metiter.c", "metiter.h" e "main.c", *e eventuais códigos de laboratórios passados usados na solução*) deve ser enviado via página da disciplina no EAD até terça da semana que vem, dia 4 de Novembro.