

INF1608 – Análise Numérica

Lab 10: Métodos Iterativos para Sistemas Lineares

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

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

1. Para a solução de sistemas lineares na forma $A\mathbf{x} = \mathbf{b}$, pede-se:

- (a) Implemente o método iterativo de Jacobi:

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

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 diferença das duas últimas iterações tiver valor menor ou igual à tolerância. A função deve armazenar a solução final em x e retornar o número de iterações efetuado. Se a matrix A não for *estritamente diagonal dominante*, a função deve retornar zero. O protótipo da função é dado por:

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

- (b) Seguindo a mesma interface, implemente o método iterativo de Gauss-Seidel:

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

O protótipo da função é dado por:

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

- (c) Novamente, seguindo a mesma interface, implemente o método iterativo de Gauss-Seidel com sobre-relaxamento ($w > 1.0$). Isto é, a função deve avaliar inicialmente \mathbf{x}_{k+1} com o método de Gauss-Seidel e então ajustar \mathbf{x}_{k+1} com a fórmula do relaxamento:

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

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

```
int SOR (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. Seu programa deve exibir na tela o número de iterações e a solução encontrada para cada um dos métodos.

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \quad e \quad \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0.5 \\ -1 & 3 & -1 & 0 & 0.5 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0.5 & 0 & -1 & 3 & -1 \\ 0.5 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 1.5 \\ 1 \\ 1 \\ 1.5 \\ 2.5 \end{bmatrix}$$

Sabe-se que as soluções destes sistemas são $[1 \ 2]$ e $[1 \ 1 \ 1 \ 1 \ 1 \ 1]$, respectivamente. Em seguida, troque as linhas dos sistemas e verifique se suas funções detectam corretamente matrizes que não são estritamente diagonal dominantes.

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*) devem ser enviados via página da disciplina no EAD. O prazo final para envio sem atraso é **segunda-feira, dia 16 de novembro**. O sistema receberá trabalhos atrasados pelo prazo adicional de uma semana, com a perda de 1.0 ponto na nota.