INF1608 – Análise Numérica

# Lab 3: Sistemas Lineares

Prof. Waldemar Celes

Departamento de Informática, PUC-Rio

Para este exercício, considere a representação de matrizes implementada no primeiro laboratório do semestre. A matriz é representada por um vetor de ponteiros, onde cada elemento aponta para o vetor linha.

1. Para a solução de sistemas lineares na forma $A\mathbf{x} = \mathbf{b}$, considere a fatoração $A = LU$, onde $L$ representa a matriz triangular inferior e $U$ a matriz triangular superior. A matrix $U$ é obtida através do procedimento de Eliminação de Gauss da matriz $A$. Para melhorar a estabilidade numérica do método, deve-se empregar a estratégia de *pivotamento*, isto é, as linhas da matriz são trocadas para garantir que o elemento pivô da eliminação de cada coluna seja sempre o elemento de maior valor absoluto da coluna em questão. As trocas devem ser registradas num vetor de permutação $\mathbf{p}$; este vetor é inicializado com `p[i] = i`; a cada troca de linhas da matriz, troca-se os elementos desse vetor. A matriz $L$ tem os elementos da diagonal iguais a 1 e os valores abaixo da diagonal iguais aos *fatores* ($f_{ij} = a_{ij}/ajj$) usados na Eliminação de Gauss da matriz $A$.

   Com a matriz fatorada, pode-se resolver o sistema linear para um vetor independente $\mathbf{b}$ qualquer, em ordem $O(n^2)$, usando substituições progressiva (*top-down*) e regressiva (*bottom-up*).

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ LU\mathbf{x} &= \mathbf{b}' \\ L\mathbf{y} &= \mathbf{b}', \text{onde } \ \mathbf{y} = U\mathbf{x} \quad e \quad \mathbf{b}'_i = \mathbf{b}_{\mathbf{p}_i} \end{aligned}$$

   Assim, dada a matriz fatorada $LU$, acha-se $\mathbf{y}$ com substituição *top-down* (substituição progressiva): $L\mathbf{y} = \mathbf{b}' = \mathbf{b}[\mathbf{p}[i]]$, onde $\mathbf{b}' = \mathbf{b}[\mathbf{p}[i]]$ representa o vetor independente com permutação de elementos de acordo com a permutação de linhas da matriz na eliminação de Gauss. Em seguida, acha-se $\mathbf{x}$ com substituição *bottom-up* (substituição regressiva ou retro-substituição): $U\mathbf{x} = \mathbf{y}$.

   Pede-se:

   (a) Implemente uma função que receba como parâmetros uma matriz quadrada $A$, de dimensão $n \times n$, e faça a fatoração $A = LU$. As matrizes $L$ e $U$ devem ser armazenadas no espaço de memória da matriz $A$, sobrescrevendo os valores dos elementos de $A$. A diagonal da matriz $L$ (com valores iguais a 1) não é armazenada. A função deve preencher o vetor de permutação $\mathbf{p}$, já alocado, também recebido como parâmetro pela função, de dimensão $n$. O protótipo da função deve ser:

```
void fatoracao (int n, double** A, int* p);
```

(b) Implemente uma função que receba como entrada a matriz $A$ fatorada (isto é, a matriz $LU$), de dimensão $n \times n$, o vetor de permutação $\mathbf{p}$ e um vetor independente $\mathbf{b}$, e preencha o vetor solução $\mathbf{x}$, já alocado, também recebido como parâmetro. Lembre-se que a matriz fatorada deve operar sobre o vetor independente permutado, isto é, `b[p[i]]`. O protótipo da função é dado por:

```
void substituicao (int n, double** A, int* p, double* b, double* x);
```

(c) Implemente uma função que resolva um sistema linear. A função deve receber como parâmetros a matriz $A$ original, de dimensão $n \times n$, e um vetor independente $\mathbf{b}$, e preencher o vetor solução $\mathbf{x}$, já alocado, também recebido como parâmetro. Essa função deve usar, obrigatoriamente, as funções dos itens anteriores, alocando e, depois liberando, eventuais estruturas auxiliares. O protótipo da função é dado por:

```
void gauss (int n, double** A, double* b, double* x);
```

2. Escreva um código para testar sua função. Você pode usar para testes iniciais os sistemas indicados abaixo:

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -6 \end{bmatrix} \quad e \quad \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0.5 \\ -1 & 3 & -1 & 0 & 0.5 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0.5 & 0 & -1 & 3 & -1 \\ 0.5 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 1.5 \\ 1.0 \\ 1.0 \\ 1.5 \\ 2.5 \end{bmatrix}$$

Sabe-se que as soluções destes sistemas são [3 1 2] e [1 1 1 1 1 1], respectivamente.

Agrupe os protótipos das funções em um módulo “sistlinear.h” e as implementações em um módulo “sistlinear.c”. Escreva um outro módulo “main.c”para testar sua implementação.

**Entrega:** O código fonte deste trabalho (isto é, os arquivos “sistlinear.h”, “sistlinear.c” e “main.c”) deve ser enviado via página da disciplina no EAD. Deve ser enviado também o código fonte do Lab 0, “matriz.h” e matriz.c”, se usado na solução. O prazo final para envio é **domingo, dia 5 de setembro**.