

Lab 6: Fatoração QR

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

Para este exercício, considere a representação de matrizes por vetor de ponteiros do Lab 0. Se usar o módulo, siga **exatamente** a interface em “matriz.h”. Não envie suas implementações desses códigos para a correção; envie apenas o código pedido deste laboratório.

Podemos resolver um sistema inconsistente na forma $A_{m \times n} \mathbf{x}_n = \mathbf{b}_m$ através da Fatoração QR.

$$A_{m \times n} = Q_{m \times n} R_{n \times n}$$

onde Q é uma matriz ortogonal e R é uma matriz triangular superior.

Na sua forma mais simples, a Fatoração QR pode ser conseguida pelo algoritmo de ortogonalização de Gram-Schmidt. A versão *modificada* que apresenta melhor estabilidade numérica é mostrada no pseudo-código abaixo:

```

for  $j = 0$  to  $n - 1$ 
     $\mathbf{w} = \mathbf{v}_j$ 
    for  $i = 0$  to  $j - 1$ 
         $r_{ij} = \mathbf{q}_i^T \mathbf{w}$ 
         $\mathbf{w} = \mathbf{w} - r_{ij} \mathbf{q}_i$ 
     $r_{jj} = \|\mathbf{w}\|$ 
     $\mathbf{q}_j = \frac{\mathbf{w}}{r_{jj}}$ 

```

De posse de Q e R , podemos fazer:

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ QR\mathbf{x} &= \mathbf{b} \\ R\mathbf{x} &= Q^T \mathbf{b} \end{aligned}$$

calculando assim, $Q^T \mathbf{b}$ e achando \mathbf{x} por retro-substituição.

1. Pede-se:

- (a) Implemente a fatoração QR por ortogonalização de Gram-Schmidt modificado. Considere que a função recebe as matrizes $Q_{m \times n}$ e $R_{n \times n}$ previamente alocadas, tendo apenas que preenchê-las, seguindo o protótipo:

```
void QR (int m, int n, double** A, double** Q, double** R);
```

- (b) Usando a função do item anterior, resolva um sistema inconsistente na forma $A_{m \times n} \mathbf{x}_n = \mathbf{b}_m$ pelo Método dos Mínimos Quadrados via Fatoração QR. Sua função deve receber como parâmetros a matriz A e o vetor \mathbf{b} , e preencher o vetor solução \mathbf{x} , previamente alocado, seguindo o protótipo:

```
double mmqQR (int m, int n, double** A, double* b, double* x);
```

2. Para testar seu código, escreva um programa com as seguintes implementações:

- (a) Resolva os sistemas inconsistentes abaixo usando o MMQ via Fatoração QR. Exiba na tela o vetor que representa a solução aproximada de cada sistema.

$$\text{a) } \begin{bmatrix} 3 & -1 & 2 \\ 4 & 1 & 0 \\ -3 & 2 & 1 \\ 1 & 1 & 5 \\ -2 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ -5 \\ 15 \\ 0 \end{bmatrix} \quad \text{b) } \begin{bmatrix} 4 & 2 & 3 & 0 \\ -2 & 3 & -1 & 1 \\ 1 & 3 & -4 & 2 \\ 1 & 0 & 1 & -1 \\ 3 & 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}$$

Agrupe os protótipos das funções pedidas em um módulo “qr.h” e as implementações em um módulo “qr.c”. Escreva um outro módulo “main.c” para o código de teste da sua implementação.

Entrega: O código fonte deste trabalho (isto é, os arquivos “qr.h”, “qr.c” e “main.c”, apenas) devem ser enviados via página da disciplina no EAD. O prazo final para envio é **quinta-feira, dia 27 de setembro**.