ÁLGEBRA LINEAR ALGORÍTMICA–PLE–LABORATÓRIO 4

1. Neste laboratório implementaremos a decomposição QR de uma matriz usando o algoritmo de Gram-Schimdt. A função `qr_decomp` deve receber como entrada uma matriz quadrada A de tamanho $n \times n$ e retornar uma matriz ortogonal Q e uma matriz triangular superior R, ambas $n \times n$, tais que $A = QR$. No resto do laboratório denotaremos por A_i e Q_i as i-ésimas colunas das matrizes A e Q.

2. A função `qr_decomp` basicamente executa o algoritmo de Gram-Schimdt com algumas particularidades, que são apontadas a seguir:

1. os vetores da base à qual será aplicada o algoritmo de Gram-Schimdt são as colnas de A;
2. à medida que forem sendo gerados, os vetores da base ortonormal serão guardados como colunas da matriz Q;
3. a entrada $r_{i,j}$ de R deve ser preenchida conforme a tabela abaixo:

Condição	Conteúdo da entrada $r_{i,j}$
$j > i$	produto interno de A_j com Q_i
$j = i$	norma do vetor A_j
$j < i$	zero

4. o cálculo dos valores de $r_{i,j}$, em uma iteração do algoritmo de Gram-Schimdt, deve ser feito usando produtos de matrizes.

⚠ Os valores não nulos da segunda coluna da tabela serão todos gerados como parte das iterações do algoritmo de Gram-Schimdt e o preenchimento das posições da matriz R deve ser feito à medida que esses valores forem sendo gerados.

Desafio. *Você consegue calcular todos os coeficientes $r_{i,j}$ em uma iteração do algoritmo de Gram-Schimdt fazendo apenas duas multiplicações de matrizes, não importando o tamanho da matriz A?*

3. A segunda função a ser implementada como parte deste laboratório vai chamar-se `qr_resolve`. Tendo como entrada uma matriz quadrada A de tamanho $n \times n$ e uma matriz coluna b de tamanho $n \times 1$, a função usa a decomposição QR de A para resolver o sistema $AX = b$, em que X é uma matriz de variáveis $n \times 1$. A função deve executar os seguintes passos:

Passo 1: aplicar `qr_decomp` à matriz A do sistema;

Passo 2: usar a saída do passo 1 para obter matrizes A_1 e b_1 de um sistema triangular superior $A_1X = b_1$ equivalente a $AX = b$;

Passo 3: gerar o vetor coluna X cujas entradas são as variáveis $x_1, \ldots, x_n$;

Passo 4: calcular a lista de equações do sistema $A_1X = b_1$;

Passo 5: resolver o sistema encontrado no passo 4.

⚠ Dois comentários importantes:

1. Mais detalhes sobre como executar o passo 2 podem ser encontradas no artigo 5.4 do capítulo 5 das notas de aula.
2. Para simplificar o passo 5, você pode usar a função `solve` do Maxima para resolver o sistema triangular.

4. A tabela abaixo lista algumas funções do Maxima que podem ser úteis na execução deste laboratório. O fato de uma função estar na tabela não significa que ela tem que aparecer nas suas implementações.

Função	Retorna
`zerofor(A)`	matriz de zeros com as mesmas dimensões da matriz A
`col(M, i)`	i-ésima coluna da matriz M
`transpose(A)`	transposta da matriz A
`ev(leq,ls)`	aplica às equações da lista `leq` as igualdades da lista `ls`, que devem ser da forma `variável = constante`
`mat_norm(u,frobenius)`	norma do vetor u representado como matriz coluna
`addcol(M,r)`	acrescenta a coluna `r` à matriz M, (`r` pode ser uma lista ou uma matriz coluna
`length(A)`	dimensão da matriz quadrada A
`solve(leq,lv)`	resolve o sistema dado pela lista de equações `leq` nas variáveis `lv`.