ÁLGEBRA LINEAR ALGORÍTMICA–2020.2–LABORATÓRIO 5

1. Neste laboratório implementaremos uma função que projeta um hipercubo do $\mathbb{R}^4$ em um subespaço de dimensão 3. A descrição abaixo pressupõe que você tenha lido com cuidado o artigo 1.3 do capítulo 6 das notas de aula. Começaremos descrevendo duas funções auxiliares.

⚠ Todas as funções neste laoratório devem ser implementadas partindo do princípio de que os vetores são representados como matrizes coluna 4×1.

2. A primeira função auxiliar de que precisaremos é `conta_diferencas` que, tendo como entrada dois vetores do $\mathbb{R}^4$, conta quantas entradas distintas estes dois vetores têm. E não há muito mais o que dizer sobre ela.

3. A segunda função auxiliar, chamada de `ortogonal`, calcula uma base ortonormal do vetor dado como entrada. A maneira mais direta de fazer isto no Maxima consiste em:

- calcular o complemento ortogonal H de v usando `orthogonal_complement(v)`;
- calcular uma base ortogonal β de H usando `gramschmidt`;
- normalizar os vetores de β para obter uma base ortonormal de H.

⚠ Quando aplicamos a função `ortogonal_complement` ao vetor

$$v : \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

o Maxima retorna

$$\text{span}\left(\begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}\right)$$

Para ter acesso aos três vetores desta base do complemento ortogonal de v precisamos usar a função `args`. Assim, atribuindo a saída de `ortogonal_complement(v)`, ilustrada acima, a uma variável `comp_ort`, teremos que `args(comp_ort)` retorna a lista de vetores

$$[\begin{pmatrix}-1\\1\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\-1\\1\end{pmatrix},\begin{pmatrix}0\\1\\-1\\0\end{pmatrix}]$$

4. A função principal, que vamos chamar de `hipercubo` terá como entrada um vetor não nulo $v \in \mathbb{R}^4$ e como saída o desenho da projeção do hipercubo no hiperplano H ortogonal a v. O código da função deve seguir as seguintes etapas:

Etapa 1: gere uma matriz V cujas colunas são os vértices do hipercubo;

Etapa 2: use a função `ortogonal` para obter uma base ortonormal β de H;

Etapa 3: construa a matriz Q cujas *linhas* são os vetores de β;

Etapa 4: gere a lista `lados` cujas entradas codificam os objetos que vão permitir que a biblioteca `draw` desenhe os vários lados da projeção do hipercubo;

Etapa 5: desenhe o hipercubo usando `draw3d` ou `wxdraw3d`.

A etapa 4 é a mais complicada e é analisada em mais detalhes abaixo. Antes, porém, duas sugestões:

(a) Use `addrow` para montar a matriz V linha a linha: a primeira linha começa com 8 zeros seguidos de 8 uns, a segunda é formada pois dois blocos de 4 zeros seguidos de 4 uns, e assim por diante.
(b) Use `addcol` para montar a transposta de Q a partir dos vetores de β.

5. Para a etapa 4, comece inicializando `lados` com a lista vazia. Em seguida, para cada par de índices

$$1 \leq i < j \leq 16$$

verifique se os vértices correspondentes às colunas i e j de V só diferem em uma entrada. Caso a resposta seja sim:

1. calcule as projeções $w_i = \mathrm{P}_H(v_i)$ e $w_j = \mathrm{P}_H(v_j)$, em que v_i e v_j são as colunas i e j de V, sobre o hiperplano H;
2. defina `vt:t·w_i+(1−t)·w_j`;
3. inclua `parametric(vt[1,1],vt[2,1],vt[3,1],t,0,1)` na lista `lados`.

Lembre-se que H é o hiperplano ortogonal ao vetor v dado como entrada. Quando t varia entre 0 e 1, a fórmula no item 2 produz os pontos do segmento de reta que liga v_i a v_j e o objeto no item 3 desenha este segmento no Maxima.

⚠ Cuidado com esta etapa: as **projeções** $\mathrm{P}_H(v_i)$ e $\mathrm{P}_H(v_j)$ devem ser ligadas quando v_i e v_j (e não suas projeções!) diferem em apenas uma de suas entradas.

5. Alguns lembretes importantes:

(a) comece as funções pondo as variáveis locais entre colchetes, caso contrário elas se tornarão globais, com consequências impredizíveis;

(b) ao usar a função `gramschmidt` do Maxima para ortogonalizar os vetores, lembre-se que ela retorna vetores que *não estão normalizados*, você vai precisar aplicar `uvect` para normalizá-los;

(c) `addrow`($\mathtt{v}_1, \mathtt{v}_2, \ldots, \mathtt{v}_s$) retorna a matriz cujas linhas são $\mathtt{v}_1, \mathtt{v}_2, \ldots, \mathtt{v}_s$;

(d) `addcol`($\mathtt{v}_1, \mathtt{v}_2, \ldots, \mathtt{v}_s$) retorna a matriz cujas colunas são $\mathtt{v}_1, \mathtt{v}_2, \ldots, \mathtt{v}_s$;

(e) o símbolo para diferente no Maxima é `#`.

⚠ Em (c) e (d) as linhas e colunas devem ser representadas, respectivamente, por matrizes linha e matrizes coluna.