ÁLGEBRA LINEAR ALGORÍTMICA–PLE–LABORATÓRIO 5

1. Neste laboratório implementaremos uma função que projeta um hipercuno do $\mathbb{R}^4$ em um subespaço de dimensão 3 e desenha o objeto resultante. A descrição abaixo pressupõe que você tenha lido com cuidado o artigo 1.3 do capítulo 6 das notas de aula.

2. Na implementação da função principal precisaremos de uma função que conta quantas entradas distintas dois vetores do $\mathbb{R}^4$ têm. A maneira mais fácil de implementar esta função é subtrair um vetor do outro e contar quantas posições são nulas. Comece implementando esta função, partindo do princípio de que os vetores serão representados como matrizes coluna 4×1.

3. A função principal, que vamos chamar de `hipercubo` terá como entrada uma lista L de três vetores linearmente independentes do $\mathbb{R}^4$ e como saída o desenho da projeção do hipercubo no espaço gerado por estes vetores. O código da função deve seguir as seguintes etapas:

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

Etapa 2: obtenha uma base ortonormal β do subespaço S gerado pelos vetores de L;

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 maneira mais fácil de gerar a matriz V é, provavelmente, montá-la 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. A etapa 4 é a mais complicada e é analisada em mais detalhes abaixo.

4. 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 v_i e v_j das colunas i e j de V no subespaço S;
2. defina $\mathtt{vt}:\mathtt{t}\cdot\mathtt{v_i}+(1-\mathtt{t})\cdot\mathtt{v_j}$;
3. inclua `parametric(vt[1,1],vt[2,1],vt[3,1],t,0,1)` na lista `lados`.

Lembre-se que S é o subespaço gerado pelos vetores da lista L. 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.

5. Alguns lembretes importantes:

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

(b) para usar as funções de álgebra linear do Maxima é necessário carregar as bibliotecas `linearalgebra` e `eigen` e, para desenhar, é preciso carregar `draw`;

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

(d) a função `addrow(A,r)` acrescenta a linha `r` ao final da matriz `A`;

(e) `col(A,i)` retorna a i-ésima linha da matriz `A`;

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