ÁLGEBRA LINEAR ALGORÍTMICA–2020.2–LABORATÓRIO 2

1. Neste laboratório investigaremos como utilizar operadores lineares para transformar quadrados de lado um de maneira a produzir figuras variadas. Mais precisamente, começando do quadrado Q de vértices

$$(0,0),(1,0),(0,1) \text{ e } \quad (1,1)$$

utilizaremos operadores lineares e translações para desenhar várias letras. Finalmente, usaremos o que aprendemos no laboratório anterior para criar um animação que faz as letras aparecerem, cada uma de uma vez, na janela gráfica do Maxima.

2. Por exemplo, para gerar a letra I, esticamos o quadrado Q ao longo da vertical aplicando aos vetores que definem cada um de seus vértices o operador cuja matriz relativa à base canônica é

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

e sem efetuar nenhum translação. O resultado é a imagem na figura 1, na qual o "I" ficou um pouco largo demais, além de não estar no centro da figura. Podemos melhorar isto estreitando o quadrado ao longo da horizontal e transladando a figura; para isto usamos a matriz

$$M_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 4 \end{bmatrix} \text{ e o vetor } \quad u = \begin{bmatrix} 0 \\ -2 \end{bmatrix}.$$

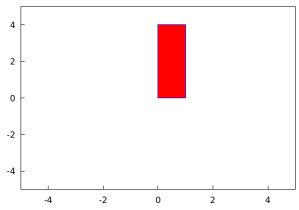

FIGURA 1. Usando só M_1

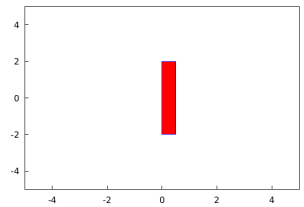

FIGURA 2. Usando M_2 e transladando

Para gerar letras mais complexas precisaremos aplicar tantos operadores e translações ao quadrado Q quantos são os vários segmentos de que a figura é constituída. Por exemplo, no caso da letra "A" são necessárias três matrizes e três translações.

3. Antes de podermos desenhar as letras, precisamos criar a função `letra(lm,lu)` cujas entradas são uma lista de matrizes `lm` e uma lista de vetores `lu`. Para cada

$$k = 1, \ldots, \texttt{length(lm)}$$

a função gera o retângulo obtido aplicando aos vértices do quadrado Q a matriz `lm[k]` e transladando a figura usando o vetor `lu[k]` como ilustrado no caso do 'Ì'. Cada retângulo é, então, posto em uma lista para que todos sejam desenhados simultaneamente. O esboço do código da função `letra` é apresentado a seguir.

```
letra(lm,lu):=
block(
    liste as variáveis locais à função,
    crie uma matriz quad cujas colunas são os vértices do quadrado Q,
    crie uma lista vazia lret,
    for k:1 thru length(lm) do
    (
      crie uma matriz v cujas 4 colunas são iguais a lu[k],
      ret:lm[k].quad+v,
      crie a lista lx das abscissas dos pontos de quad,
      crie a lista ly das ordenadas dos pontos de quad,
      retangulo:polygon(lx,ly),
      acrescente retangulo à lista lret
    ),
  retorne lret
)$
```

Note que, neste código, as listas `lm` e `lu` têm que ter a mesma quantidade de entradas. A lista vazia é `[]`.

4. Além das funções do Maxima que já apareceram no laboratório 1, você precisará também das que estão listadas na tabela 1. Duas observações importantes sobre estas funções:

1. `cons(a,l)` acrescenta um novo elemento `a` no início da lista `l`, *mas não atribui esta lista à variável* `l`. Se o que você quer é uma lista, que continua sendo chamada de `l`, mas que agora é encabeçada por um novo elemento `a`, você tem que atribuir `cons(a,l)` a `l`, fazendo `l:cons(a,l)`.
2. É necessário um pouco de cuidado no uso da função `polygon(lx,ly)`. Em primeiro lugar, a j-ésima posição nas listas `lx` e `ly` deve conter a abscissa e a ordenada *de um mesmo vértice* do polígono. Em segundo lugar, os vértices devem estar listados em `lx` e `ly` na ordem em que os pontos seriam ligados se estivéssemos desenhado o polígono *sem tirar o lápis do papel*. Por exemplo, `polygon(lx,ly)` gera um quadrado se `lx:[0,1,1,0]` e `ly:[0,0,1,1]`, mas não se `lx:[0,1,1,0]` e `ly:[0,1,0,1]`. Vale a pena experimentar este segundo caso para ver o que acontece.

Função	Efeito
`l[k]`	retorna o k-ésimo elemento da lista `l`
`polygon(lx,ly)`	desenha o polígono cujos vértices têm abscissas em `lx` e ordenadas em `ly`
`length(l)`	retorna o tamanho da lista `l`
`cons(a,l)`	acrescenta o elemento `a` à lista `l`

TABELA 1. Funções usadas no código de `letra`.

5. Por exemplo, o código abaixo gera a letra "A" ilustrada na figura 3.

```
D1:matrix([2,0],[0,1/2]);/*Cria uma barra longa e estreita */
A1:matrix([1,0],[1,1]);/*Cisalhamento vertical inclina a barra para cima*/
A1:A1.D1;
R1:matrix([-1,0],[0,1]);/*Reflete em torno do eixo vertical*/
A2:R1.A1;
R2:matrix([1,0],[0,-1]);/*Reflete em torno do eixo horizontal*/
D2:matrix([1,0],[0,4]); /*Estica ao longo da vertical*/
A3:D2.R2.A1;
A4:D2.R2.A2;
A5:matrix([2.5,0],[0,-1/2]);
lm:[A3,A4,A5];
lu:[[0,3.5],[0,3.5],[-1.3,-0.3]];
wdraw2d(xrange=[-5,5],yrange=[-5,5],letra(lm,lu));
```

Antes de executar o código acima para desenhar o "A", voce precisará carregar a biblioteca gráfica do Maxima executando `load(draw)`. Em algumas instalações do Maxima é necessário pôr a palavra `draw` entre aspas duplas.

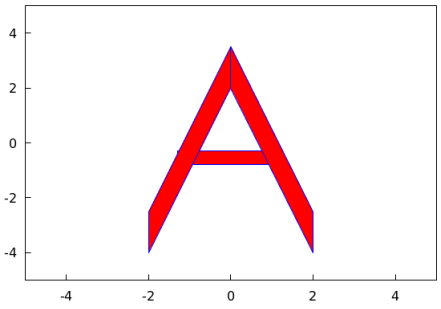

FIGURA 3. Letra A gerada pelo Maxima

6. Invente listas de matrizes `lm` e de vetores `lu` que, quando tomadas como entrada da função `letra` fazem o programa retornar cada uma das letras maiúsculas:

I, L, V, F, H, M, e S em caracteres romanos e I, F em itálico.

Para cada letra você vai precisar de tantas matrizes e vetores quantos são os segmentos que formam a letra. Como usaremos estas listas de matrizes em uma animação, todas as letras devem ser desenhadas relativamente a um mesmo tamanho de janela gráfica, definido usando `xrange` e `yrange`.

7. Finalmente, utilizaremos `with_slider_draw` para animar as figuras, de modo que as letras apareçam na janela gráfica uma de cada vez. A maneira como fiz isto foi criando duas listas, que chamei de `lms` e `lus`, cujos elementos são as listas `lm` e `lu` necessárias para gerar as várias letras. Assim, posso gerar as letras usando `letra(lms[i],lus[i])`, em que `i` é um índice que varia de 1 a 9, já que devemos gerar nove letras.

Não esqueça que o nome do seu arquivo deve ter a forma `nome_DRE_lab02`. Além disso, seu nome completo e DRE devem constar de uma célula de texto no início do arquivo.