Respostas da Lista 5

MAT 038 (GAAL)

22/6/2016

1 Respostas da Lista de Exercícios 5

1.1 Espaços Rn

http://bit.ly/mat038-ta2 e http://bit.ly/mat038-tb2

Exercício 1

(a)

```
A = matrix(QQ, [[5, 0, -10, 10], [-3, 4, 18, -2], [1, 3, 7, 5]])
A
```

```
[  5   0 -10  10]
[ -3   4  18  -2]
[  1   3   7   5]
```

```
A.echelon_form()
```

```
[ 1  0 -2  2]
[ 0  1  3  1]
[ 0  0  0  0]
```

Logo o sistema tem soluções diferentes de zero.

Em particular $2V_1 + V_2 = X$.

(b)

```
A = matrix(QQ, [[5, 0, -10, -2], [-3, 4, 18, -1], [1, 3, 7, 1]])
A
```

```
[  5   0 -10  -2]
[ -3   4  18  -1]
[  1   3   7   1]
```

```
A.echelon_form()
```
```
[ 1  0 -2  0]
[ 0  1  3  0]
[ 0  0  0  1]
```

Logo o sistema não tem solução. Portanto X não é uma combinação linear de V_1, V_2 e V_3.

Exercício 2

(a)

```
A = matrix(QQ, [[1, 1, 4, 0], [1, 0 ,6, 0], [2, 0, 12, 0]])
A
```
```
[ 1  1  4  0]
[ 1  0  6  0]
[ 2  0 12  0]
```

```
A.echelon_form()
```
```
[ 1  0  6  0]
[ 0  1 -2  0]
[ 0  0  0  0]
```

Logo o sistema tem um número infinito de soluções. Portanto V_1, V_2 e V_3 são linearmente dependentes.

(b)

```
A = matrix(QQ, [[1, 2, 3, 0], [1, 3, 1, 0], [1, 1, 2, 0]])
A
```
```
[1 2 3 0]
[1 3 1 0]
[1 1 2 0]
```

```
A.echelon_form()
```
```
[1 0 0 0]
[0 1 0 0]
[0 0 1 0]
```

Logo a única solução do sistema é c_1 = 0, c_2 = 0 e c_3 = 0. Portanto V_1, V_2 e V_3 são linearmente independentes.

Exercício 3

Os dois vetores são linearmente dependentes e somente se um vetor é um múltiplo não-nulo do outro, ou seja, V_1 = c V_2. Dessa forma determinamos o valor de c e depois disso o valor de lambda. O valor de lambda deve ser igual a 2 ou -2.

Exercício 4

Escolha um ponto P_1 que pertence à reta 1 e um ponto P_2 que pertence à reta 2. Considere os vetores diretores V_1 e V_2 das retas 1 e 2, respectivamente. Para que as retas 1 e 2 pertençam a um mesmo plano, os vetores $\overrightarrow{P_1P_2}$, V_1 e V_2 devem ser linearmente dependentes. Isso ocorre se o produto misto desses três vetores for igual a zero. Isso impõe uma condição sobre a constante a.

Exercício 5

```
A = matrix([[1, 1, -2], [-1, 2, 1], [0, 1, -1]])
A
```

```
[ 1  1 -2]
[-1  2  1]
[ 0  1 -1]
```

```
A.eigenvalues()
```

```
[2, 1, -1]
```

```
A.eigenvectors_right()
```

```
[(2, [
(1, 3, 1)
], 1), (1, [
(1, 2/3, 1/3)
], 1), (-1, [
(1, 0, 1)
], 1)]
```

Portanto, o vetor (1, 3, 1) forma uma base para o auto-subespaço associado ao autovetor 2, o vetor (1, 2/3, 1/3) forma uma base para o auto-subespaço associado ao autovalor 1, e o vetor (1, 0, 1) forma uma base para o auto-subespaço associado ao autovalor -1.

Exercício 6

```
v1 = vector([4, 2, -3])
v2 = vector([2, 1, -2])
v3 = vector([-2, -1, 0])
```

(a)

```
v1.cross_product(v2).dot_product(v3)
```

```
0
```

(b)

```
v1.cross_product(v2)
```

```
(-1, 2, 0)
```

```
v1.cross_product(v2) != vector([0, 0, 0])
```

```
True
```

(c)

A dimensão do subespaço é igual a 2

(d)

O subespaço gerado por v1 v2 e v3 é um plano que passa pela origem.

Exercício 7

```
v1 = vector([2, 1, 3])
v2 = vector([2, 6, 4])
```

(a)

Não pois esse conjunto tem dimensão no máximo igual a 2 e R3 tem dimensão igual a 3.

(b)

```
v3 = v1.cross_product(v2)
v3
```

```
(-14, -2, 10)
```

Com essa escolha v1, v2 e v3 formam uma base para R3 pois eles formam um conjunto com 3 vetores linearmente independentes. De fato

```
v1.cross_product(v2).dot_product(v3) != 0
```

```
True
```

Exercício 8

```
v1 = vector([1, 1, -1, 0])
v2 = vector([0, 2, 0, 1])
v3 = vector([-1, 0, 0, 1])
```

```
w1 = v1
w2 = v2 - (v2.dot_product(w1)/w1.norm()^2)*w1
w3 = v3 - (v3.dot_product(w1)/w1.norm()^2)*w1 - (v3.dot_product(w2)/\
   w2.norm()^2)*w2
u1 = w1/w1.norm()
u2 = w2/w2.norm()
u3 = w3/w3.norm()
```

```
u1
u2
u3
```

```
(1/3*sqrt(3), 1/3*sqrt(3), -1/3*sqrt(3), 0)
(-2/11*sqrt(11/3), 4/11*sqrt(11/3), 2/11*sqrt(11/3), 3/11*sqrt(11/3))
```

```
(-2/5*sqrt(10/11), -3/10*sqrt(10/11), -7/10*sqrt(10/11), 3/5*sqrt(10/11))
```

```
u1.norm()
u2.norm()
u3.norm()
```

```
1
1
1
```

```
u1.dot_product(u2)
u1.dot_product(u3)
u2.dot_product(u3)
```

```
0
0
0
```

Exercício 9

(a)

```
u1 = vector([1/sqrt(2), -1/sqrt(2)])
u2 = vector([1/sqrt(2), 1/sqrt(2)])
p = vector([1, 3])
```

```
u1.norm()
u2.norm()
u1.dot_product(u2)
```

```
1
1
0
```

```
Q = matrix([[1/sqrt(2), 1/sqrt(2)], [-1/sqrt(2), 1/sqrt(2)]])
Q
```

```
[ 1/2*sqrt(2)  1/2*sqrt(2)]
[-1/2*sqrt(2)  1/2*sqrt(2)]
```

```
Q.transpose() == Q.inverse()
```

```
True
```

```
p_novo = Q.transpose()*p
p_novo
```

```
(-sqrt(2), 2*sqrt(2))
```

(b)

```
u1 = vector([1/sqrt(2), -1/sqrt(2), 0])
u2 = vector([0, 0, 1])
u3 = vector([1/sqrt(2), 1/sqrt(2), 0])
```

```
p = vector([2, -1, 3])
```

```
u1.norm()
u2.norm()
u3.norm()
```

```
1
1
1
```

```
u1.dot_product(u2)
u1.dot_product(u3)
u3.dot_product(u2)
```

```
0
0
0
```

```
Q = matrix([[1/sqrt(2), 0, 1/sqrt(2)], [-1/sqrt(2), 0, 1/sqrt(2)], \
    [0, 1, 0]])
Q
```

```
[ 1/2*sqrt(2)           0  1/2*sqrt(2)]
[-1/2*sqrt(2)           0  1/2*sqrt(2)]
[          0            1            0]
```

```
Q.transpose() == Q.inverse()
```

```
True
```

```
p_novo = Q.transpose()*p
p_novo
```

```
(3/2*sqrt(2), 3, 1/2*sqrt(2))
```

Exercício 10

(a)

Sim. Por exemplo, se n = 3, S_1 = (1, 0, 0), (2, 0, 0) e S_2 = (1, 0, 0), (2, 0, 0), (0, 1, 0).

(b)

Sim. Por exemplo, se n = 3, S_1 = (2, 0, 0), (0, 1, 0) e S_2 = (1, 0, 0), (2, 0, 0), (0, 1, 0).

Exercício 11

```
A = matrix([[0, 0, 0], [0, 2, 2], [0, 2, 2]])
A
```

```
[0 0 0]
[0 2 2]
[0 2 2]
```

```
A.eigenvalues()
```

```
[4, 0, 0]
```

```
A.eigenvectors_right()
```

```
[(4, [
(0, 1, 1)
], 1), (0, [
(1, 0, 0),
(0, 1, -1)
], 2)]
```

Portanto (0, 1, 1) forma uma base para o auto-subespaço associado ao autovalor 4 e (1, 0, 0) e (0, 1, -1) formam uma base para o auto-subespaço associado ao autovalor 0.