

Aula Prática 5 (ALN)

Raphael F. Levy

June 8, 2021

Introdução

Para a Aula Prática 5, os métodos estudados foram os de decomposição QR , usados para decompor uma matriz A em uma ortogonal Q e uma triangular superior R . Aqui, usamos três diferentes métodos: o Método de Gram-Schmidt, que foi modificado com e sem pivoteamento de colunas, o Método de Householder e o Algoritmo QR em si, para encontrar os autovalores de uma matriz A simétrica.

1 Método de Gram-Schmidt

Para a questão 1, foi desenvolvido uma função para calcular o Método de Gram-Schmidt, que pode ser encontrado no arquivo *Metodo_Gram_Schmidt.sci*.

```
1 --> Q=[-0.8 0.546; 0.6 0.728; 0 -0.414]
2 Q =
3
4     -0.8    0.546
5     0.6    0.728
6     0.    -0.414
7
8 --> R=[5 0.4; 0 2.417]
9 R =
10
11     5.    0.4
12     0.    2.417
13
14 --> A=Q*R
15 A =
16
17    -4.    0.999682
18     3.    1.999576
19     0.   -1.000638
20
21 --> [Q,R]=qr_GS(A)
22 Q =
23
24    -0.8    0.5461376
25     0.6    0.7281835
26     0.   -0.4141044
```

```

27 R =
28
29     5.    0.4
30     0.    2.4163908
31
32 --> Q';
33
34 --> QtQ=Q'*Q
35 QtQ =
36
37     1.          5.551D-17
38     5.551D-17    1.
39
40 --> I=eye(2,2)
41 I =
42
43     1.    0.
44     0.    1.
45
46 --> norm(QtQ-I)
47 ans =
48
49     5.551D-17
50
51 --> norm(A-Q*R)
52 ans =
53
54     0.
55 -----
56
57 --> A=[4 9 -14; 3 13 2; 0 5 0]
58 A =
59
60     4.    9.   -14.
61     3.   13.    2.
62     0.    5.    0.
63
64 --> [Q,R]=qr_GS(A)
65 Q =
66
67     0.8   -0.4242641   -0.4242641
68     0.6    0.5656854    0.5656854
69     0.    0.7071068   -0.7071068
70 R =
71
72     5.    15.   -10.
73     0.    7.0710678    7.0710678
74     0.    0.    7.0710678
75
76 --> Q';
77
78 --> QtQ=Q'*Q
79 QtQ =
80
81     1.          -5.551D-17    2.220D-16
82    -5.551D-17    1.          1.015D-17
83     1.984D-16   -4.536D-17    1.

```

```

84
85 --> I=eye(3,3)
86 I =
87
88     1.    0.    0.
89     0.    1.    0.
90     0.    0.    1.
91
92 --> norm(QtQ-I)
93 ans =
94
95     2.331D-16
96
97 --> norm(A-Q*R)
98 ans =
99
100     4.555D-16
101 -----
102
103 --> A=[1 0 1 1; 0 1 1 1; 1 1 1 0; 1 1 0 1]
104 A =
105
106     1.    0.    1.    1.
107     0.    1.    1.    1.
108     1.    1.    1.    0.
109     1.    1.    0.    1.
110
111 --> [Q,R]=qr_GS(A)
112 Q =
113
114     0.5773503   -0.5163978    0.5070926    0.3779645
115     0.          0.7745967    0.5070926    0.3779645
116     0.5773503    0.2581989    0.1690309   -0.7559289
117     0.5773503    0.2581989   -0.6761234    0.3779645
118 R =
119
120     1.7320508    1.1547005    1.1547005    1.1547005
121     0.          1.2909944    0.5163978    0.5163978
122     0.          0.          1.183216    0.3380617
123     0.          0.          0.          1.1338934
124
125 --> Q';
126
127 --> QtQ=Q'*Q
128 QtQ =
129
130     1.          -2.321D-16   -2.128D-16   -1.951D-16
131   -2.321D-16    1.          1.899D-16    1.645D-16
132   -2.128D-16    1.899D-16    1.          -2.077D-17
133   -1.951D-16    1.645D-16   -2.077D-17    1.
134
135 --> I=eye(4,4)
136 I =
137
138     1.    0.    0.    0.
139     0.    1.    0.    0.
140     0.    0.    1.    0.

```

```

141     0.    0.    0.    1.
142
143 --> norm(QtQ-I)
144 ans  =
145
146     6.003D-16
147
148 --> norm(A-Q*R)
149 ans  =
150
151     1.118D-16
152 -----
153
154 --> A=[1 2 3; -1 1 1; 1 1 1; 1 1 1]
155 A  =
156
157     1.    2.    3.
158    -1.    1.    1.
159     1.    1.    1.
160     1.    1.    1.
161
162 --> [Q,R]=qr_GS(A)
163 Q  =
164
165     0.5    0.5735393    0.6488857
166    -0.5    0.8029551    -0.3244428
167     0.5    0.1147079    -0.4866643
168     0.5    0.1147079    -0.4866643
169 R  =
170
171     2.    1.5    2.
172     0.    2.1794495    2.7529888
173     0.    0.    0.6488857
174
175 --> Q';
176
177 --> QtQ=Q'*Q
178 QtQ  =
179
180     1.    0.    0.
181     0.    1.    -1.277D-15
182     0.    -1.265D-15    1.
183
184 --> I=eye(3,3)
185 I  =
186
187     1.    0.    0.
188     0.    1.    0.
189     0.    0.    1.
190
191 --> norm(QtQ-I)
192 ans  =
193
194     1.512D-15
195
196 --> norm(A-Q*R)
197 ans  =

```

198
199

0.

Observando os testes feitos, incluindo a ortogonalidade das matrizes $Q^T Q$ calculadas, podemos ver que o método é de fato preciso, já que $Q^T Q$ calculadas experimentalmente sempre foram uma matriz identidade, apenas não sendo representadas adequadamente devido à precisão do *Scilab*, e isso pode ser novamente confirmado encontrando a norma de $Q^T Q - I$, que foram praticamente 0 para todas as matrizes, assim como a norma de $A - Q * R$, que foi praticamente 0 em dois dos 4 testes e precisamente 0 nos outros dois..

2 Método de Gram-Schmidt Modificado

Para a questão 2, foi desenvolvido uma função para calcular o Método de Gram-Schmidt modificado, que pode ser encontrado no arquivo *Metodo_Gram_Schmidt_Mod.sci*.

```
1 --> Q=[-0.8 0.546; 0.6 0.728; 0 -0.414]
2 Q =
3
4 -0.8    0.546
5  0.6    0.728
6  0.    -0.414
7
8 --> R=[5 0.4; 0 2.417]
9 R =
10
11 5.    0.4
12 0.    2.417
13
14 --> A=Q*R
15 A =
16
17 -4.    0.999682
18  3.    1.999576
19  0.    -1.000638
20
21 --> [Q,R]=qr_GSM(A)
22 Q =
23
24 -0.8    0.5461376
25  0.6    0.7281835
26  0.    -0.4141044
27 R =
28
29 5.    0.4
30 0.    2.4163908
31
32 --> Q';
33
34 --> QtQ=Q'*Q
35 QtQ =
36
37 1.          5.551D-17
38 5.551D-17    1.
```

```

39
40 --> I=eye(2,2)
41 I =
42
43     1.    0.
44     0.    1.
45
46 --> norm(QtQ-I)
47 ans =
48
49     5.551D-17
50
51 --> norm(A-Q*R)
52 ans =
53
54     0.
55 -----
56
57 --> A=[4 9 -14; 3 13 2; 0 5 0]
58 A =
59
60     4.    9.   -14.
61     3.   13.    2.
62     0.    5.    0.
63
64 --> [Q,R]=qr_GSM(A)
65 Q =
66
67     0.8   -0.4242641   -0.4242641
68     0.6    0.5656854    0.5656854
69     0.    0.7071068   -0.7071068
70 R =
71
72     5.    15.       -10.
73     0.    7.0710678    7.0710678
74     0.    0.         7.0710678
75
76 --> Q';
77
78 --> QtQ=Q'*Q
79 QtQ =
80
81     1.         -5.551D-17    1.665D-16
82    -5.551D-17     1.         8.865D-17
83     1.429D-16    8.865D-17     1.
84
85 --> I=eye(3,3)
86 I =
87
88     1.    0.    0.
89     0.    1.    0.
90     0.    0.    1.
91
92 --> norm(QtQ-I)
93 ans =
94
95     2.838D-16

```

```

96
97 --> norm(A-Q*R)
98 ans =
99
100 8.882D-16
101 -----
102
103 --> A=[1 0 1 1; 0 1 1 1; 1 1 1 0; 1 1 0 1]
104 A =
105
106 1. 0. 1. 1.
107 0. 1. 1. 1.
108 1. 1. 1. 0.
109 1. 1. 0. 1.
110
111 --> [Q,R]=qr_GSM(A)
112 Q =
113
114 0.5773503 -0.5163978 0.5070926 0.3779645
115 0. 0.7745967 0.5070926 0.3779645
116 0.5773503 0.2581989 0.1690309 -0.7559289
117 0.5773503 0.2581989 -0.6761234 0.3779645
118 R =
119
120 1.7320508 1.1547005 1.1547005 1.1547005
121 0. 1.2909944 0.5163978 0.5163978
122 0. 0. 1.183216 0.3380617
123 0. 0. 0. 1.1338934
124
125 --> Q';
126
127 --> QtQ=Q'*Q
128 QtQ =
129
130 1. -2.321D-16 -2.214D-16 -1.759D-16
131 -2.321D-16 1. -8.856D-17 -7.230D-17
132 -2.214D-16 -8.856D-17 1. 6.784D-17
133 -1.759D-16 -7.230D-17 6.784D-17 1.
134
135 --> I=eye(4,4)
136 I =
137
138 1. 0. 0. 0.
139 0. 1. 0. 0.
140 0. 0. 1. 0.
141 0. 0. 0. 1.
142
143 --> norm(QtQ-I)
144 ans =
145
146 4.563D-16
147
148 --> norm(A-Q*R)
149 ans =
150
151 1.578D-16
152 -----

```

```

153 --> A=[1 2 3; -1 1 1; 1 1 1; 1 1 1]
154 A =
155
156      1.      2.      3.
157     -1.      1.      1.
158      1.      1.      1.
159      1.      1.      1.
160
161 --> [Q,R]=qr_GSM(A)
162 Q =
163
164      0.5      0.5735393      0.6488857
165     -0.5      0.8029551     -0.3244428
166      0.5      0.1147079     -0.4866643
167      0.5      0.1147079     -0.4866643
168
169 R =
170
171      2.      1.5      2.
172      0.      2.1794495      2.7529888
173      0.      0.      0.6488857
174
175 --> Q';
176
177 --> QtQ=Q'*Q
178 QtQ =
179
180      1.      0.      0.
181      0.      1.     -1.277D-15
182      0.     -1.265D-15      1.
183
184 --> I=eye(3,3)
185 I =
186
187      1.      0.      0.
188      0.      1.      0.
189      0.      0.      1.
190
191 --> norm(QtQ-I)
192 ans =
193
194      1.512D-15
195
196 --> norm(A-Q*R)
197 ans =
198
199      0.

```

Comparando $Q^T Q$ entre as matrizes nos dois métodos, é visível que ambas são sempre extremamente próximas. Contudo, é possível perceber que os valores nas matrizes do método modificado são ainda mais próximas de 0 do que fazendo o teste pelo método simples.

3 Método de Gram-Schmidt com Pivoteamento

Para a questão 3, foi desenvolvido uma função para calcular o Método de Gram-Schmidt com Pivoteamento de colunas, que pode ser encontrado no arquivo *Metodo_Gram_Schmidt_Pivot.sci*. Contudo, o método não está funcional devido a problemas no código.

4 Método de Householder

Para a questão 4, foram desenvolvidas duas funções, uma para calcular o Método de Householder e achar as matrizes U e R a partir de A , e uma para achar a matriz Q a partir da matriz U retornada, que podem ser encontradas no arquivo *Metodo_Householder.sci*.

```
1 --> A=[-4. 0.999682; 3. 1.999576; 0. -1.000638]
2 A =
3
4 -4.    0.999682
5 3.    1.999576
6 0.    -1.000638
7
8 --> [U,R]=qr_House(A)
9 U =
10
11 -0.9486833    0.
12 0.3162278    0.9772997
13 0.          -0.2118615
14 R =
15
16 5.    0.4
17 0.    -2.4163908
18 0.    0.
19
20 --> [Q]=constroi_Q_House(U)
21 Q =
22
23 -0.8    -0.5461376    0.2484626
24 0.6    -0.7281835    0.3312835
25 0.      0.4141044    0.9102294
26
27 --> Q';
28
29 --> QtQ=Q'*Q
30 QtQ =
31
32 1.          -1.665D-16    5.551D-17
33 -1.665D-16    1.          -2.601D-17
34 6.481D-17    -8.152D-17    1.
35
36 --> I=eye(3,3)
37 I =
38
39 1.    0.    0.
40 0.    1.    0.
```

```

41     0.    0.    1.
42
43 --> norm(QtQ-I)
44 ans  =
45
46     3.843D-16
47
48 --> norm(A-Q*R)
49 ans  =
50
51     1.099D-15
52 -----
53
54 --> A=[4 9 -14; 3 13 2; 0 5 0]
55 A  =
56
57     4.    9.   -14.
58     3.   13.    2.
59     0.    5.    0.
60
61 --> [U,R]=qr_House(A)
62 U  =
63
64     0.9486833    0.         0.
65     0.3162278    0.9238795    0.
66     0.          0.3826834    1.
67 R  =
68
69     -5.   -15.    10.
70     0.   -7.0710678  -7.0710678
71     0.    0.   -7.0710678
72
73 --> [Q]=constroi_Q_House(U)
74 Q  =
75
76     -0.8    0.4242641  -0.4242641
77     -0.6   -0.5656854    0.5656854
78     0.   -0.7071068  -0.7071068
79
80 --> Q';
81
82 --> QtQ=Q'*Q
83 QtQ  =
84
85     1.          1.110D-16  -1.665D-16
86     1.110D-16    1.        -1.015D-17
87     -1.429D-16  -1.015D-17    1.
88
89 --> I=eye(3,3)
90 I  =
91
92     1.    0.    0.
93     0.    1.    0.
94     0.    0.    1.
95
96 --> norm(QtQ-I)
97 ans  =

```

```

98      4.183D-16
99
100
101 --> norm(A-Q*R)
102 ans =
103
104      14.142136
105 -----
106
107 --> A=[1 0 1 1; 0 1 1 1; 1 1 1 0; 1 1 0 1]
108 A =
109
110      1.      0.      1.      1.
111      0.      1.      1.      1.
112      1.      1.      1.      0.
113      1.      1.      0.      1.
114
115 --> [U,R]=qr_House(A)
116 U =
117
118      0.8880738      0.      0.      0.
119      0.      0.9419651      0.      0.
120      0.3250576      0.2373833      -0.756429      0.
121      0.3250576      0.2373833      -0.6540758      1.
122 R =
123
124      -1.7320508      -1.1547005      -1.1547005      -1.1547005
125      0.      -1.2909944      -0.5163978      -0.5163978
126      0.      0.      1.183216      0.3380617
127      0.      0.      0.      1.1338934
128
129 --> [Q]=constroi_Q_House(U)
130 Q =
131
132      -0.5773503      0.5163978      0.5070926      -0.3779645
133      0.      -0.7745967      0.5070926      -0.3779645
134      -0.5773503      -0.2581989      0.1690309      0.7559289
135      -0.5773503      -0.2581989      -0.6761234      -0.3779645
136
137 --> Q';
138
139 --> QtQ=Q'*Q
140 QtQ =
141
142      1.      3.235D-16      2.761D-16      -2.000D-16
143      3.235D-16      1.      -5.110D-16      4.081D-16
144      2.761D-16      -5.110D-16      1.      2.188D-16
145      -2.000D-16      4.081D-16      2.188D-16      1.
146
147 --> I=eye(4,4)
148 I =
149
150      1.      0.      0.      0.
151      0.      1.      0.      0.
152      0.      0.      1.      0.
153      0.      0.      0.      1.
154

```

```

155 --> norm(QtQ-I)
156 ans =
157
158 1.758D-15
159
160 --> norm(A-U*R)
161 ans =
162
163 6.2139499
164 -----
165
166 --> A=[1 2 3; -1 1 1; 1 1 1; 1 1 1]
167 A =
168
169 1. 2. 3.
170 -1. 1. 1.
171 1. 1. 1.
172 1. 1. 1.
173
174 --> [U,R]=qr_House(A)
175 U =
176
177 0.8660254 0. 0.
178 -0.2886751 0.9985326 0.
179 0.2886751 -0.0382921 -0.9238795
180 0.2886751 -0.0382921 -0.3826834
181 R =
182
183 -2. -1.5 -2.
184 0. -2.1794495 -2.7529888
185 0. 0. 0.6488857
186 0. 0. 0.
187
188 --> [Q]=constroi_Q_House(U)
189 Q =
190
191 -0.5 -0.5735393 0.6488857 -3.514D-17
192 0.5 -0.8029551 -0.3244428 3.522D-17
193 -0.5 -0.1147079 -0.4866643 -0.7071068
194 -0.5 -0.1147079 -0.4866643 0.7071068
195
196 --> Q';
197
198 --> QtQ=Q'*Q
199 QtQ =
200
201 1. -5.667D-16 6.413D-16 -9.524D-18
202 -5.667D-16 1. 2.292D-16 -1.438D-17
203 6.413D-16 2.292D-16 1. -1.179D-16
204 -9.524D-18 -1.438D-17 -1.179D-16 1.
205
206 --> I=eye(4,4)
207 I =
208
209 1. 0. 0. 0.
210 0. 1. 0. 0.
211 0. 0. 1. 0.

```

```

212     0.    0.    0.    1.
213
214 --> norm(QtQ-I)
215 ans  =
216
217     1.739D-15
218
219 --> norm(A-Q*R)
220 ans  =
221
222     6.639D-15

```

4.1

Comparando as matrizes $Q^T Q$ s obtidas diretamente das matrizes A pelo método de Gram-Schmidt e as $Q^T Q$ s obtidas pelas matrizes U obtidas pelo método de Householder, podemos perceber que ambas são sempre bem próximas da matriz identidade. A maior diferença que pode ser notada, como esperado, é na ordem da matriz $Q^T Q$ s. Pelo método de Gram-Schmidt, se A é uma matriz $m \times n$, $Q^T Q$ s será $n \times n$, como pode ser visto na primeira e quarta matriz utilizada. Por outro lado, pelo método de Householder, já que Q é obtida por U , já que U será $m \times m$, consequentemente Q também o será.

4.2

```

1  --> A = [0.70000 0.70711; 0.70001 0.70711]
2  A  =
3
4      0.7          0.70711
5      0.70001      0.70711
6
7  --> [Q,R]=qr_GS(A)
8  Q  =
9
10     0.7071017      0.7071118
11     0.7071118     -0.7071017
12  R  =
13
14     0.9899566      1.0000046
15     0.           0.0000071
16
17 --> Q';
18
19 --> QtQ=Q'*Q
20 QtQ  =
21
22     1.           2.301D-11
23     2.301D-11      1.
24 -----
25

```

```

26 --> [Q,R]=qr_GSM(A)
27 Q =
28
29     0.7071017    0.7071118
30     0.7071118   -0.7071017
31 R =
32
33     0.9899566    1.0000046
34     0.          0.0000071
35
36 --> Q';
37
38 --> QtQ=Q'*Q
39 QtQ =
40
41     1.          2.301D-11
42     2.301D-11    1.
43 -----
44
45 --> [U,R]=qr_House(A)
46 U =
47
48     0.9238782    0.
49     0.3826867    1.
50 R =
51
52    -0.9899566   -1.0000046
53     0.          -0.0000071
54
55 --> [Q]=constroi_Q_House(U)
56 Q =
57
58    -0.7071017    0.7071118
59    -0.7071118   -0.7071017
60
61 --> Q';
62
63 --> QtQ=Q'*Q
64 QtQ =
65
66     1.    0.
67     0.    1.

```

Comparando os dois métodos de Gram-Shmidt, é possível ver que os resultados foram os mesmos, possivelmente com uma diferença em uma casa menor que o *Scilab* pode alcançar. $Q^T Q$ nesse caso também foi próximo de I , contudo, olhando logo acima é possível ver que, pelo menos para essa matriz A , o método de Householder teve um resultado extremamente preciso, já que seu $Q^T Q$ com Q obtido pela matriz U foi exatamente uma matriz identidade.

4.3

```

1 --> A=[1 2 3; 4 5 6; 7 8 7; 4 2 3; 4 2 2]
2 A =

```

```

3
4     1.   2.   3.
5     4.   5.   6.
6     7.   8.   7.
7     4.   2.   3.
8     4.   2.   2.
9
10 --> [Q,R]=qr_GS(A)
11 Q   =
12
13     0.1010153   0.3161731   0.5419969
14     0.404061    0.3533699   0.5161875
15     0.7071068    0.3905667  -0.5247906
16     0.404061   -0.5579525   0.3871406
17     0.404061   -0.5579525  -0.1204438
18 R   =
19
20     9.8994949   9.4954339   9.6974644
21     0.           3.2919196   3.0129434
22     0.           0.           1.9701157
23
24 --> [Q,R]=qr_GSM(A)
25 Q   =
26
27     0.1010153   0.3161731   0.5419969
28     0.404061    0.3533699   0.5161875
29     0.7071068    0.3905667  -0.5247906
30     0.404061   -0.5579525   0.3871406
31     0.404061   -0.5579525  -0.1204438
32 R   =
33
34     9.8994949   9.4954339   9.6974644
35     0.           3.2919196   3.0129434
36     0.           0.           1.9701157
37
38 --> [U,R]=qr_House(A)
39 U   =
40
41     0.741962     0.           0.
42     0.2722923    0.7865551    0.
43     0.4765114    0.1191972  -0.9800408
44     0.2722923   -0.4284409    0.1842055
45     0.2722923   -0.4284409  -0.0747553
46 R   =
47
48    -9.8994949   -9.4954339  -9.6974644
49     0.           -3.2919196  -3.0129434
50     0.           0.           1.9701157
51     0.           0.           0.
52     0.           0.           0.
53
54 --> [Q]=constroi_Q_House(U)
55 Q   =
56
57    -0.1010153   -0.3161731   0.5419969  -0.6842085  -0.3576711
58    -0.404061    -0.3533699   0.5161875   0.3280084   0.5812274
59    -0.7071068   -0.3905667  -0.5247906   0.0093972  -0.2682612

```

```

60 -0.404061    0.5579525    0.3871406    0.3655973    -0.4918175
61 -0.404061    0.5579525    -0.1204438    -0.5389987    0.4694651
62
63 --> [Q,R]=qr(A)
64 Q  =
65
66 -0.1010153   -0.3161731    0.5419969   -0.6842085   -0.3576711
67 -0.404061    -0.3533699    0.5161875    0.3280084    0.5812274
68 -0.7071068   -0.3905667   -0.5247906    0.0093972   -0.2682612
69 -0.404061    0.5579525    0.3871406    0.3655973   -0.4918175
70 -0.404061    0.5579525   -0.1204438   -0.5389987    0.4694651
71 R  =
72
73 -9.8994949   -9.4954339   -9.6974644
74  0.          -3.2919196   -3.0129434
75  0.           0.          1.9701157
76  0.           0.           0.
77  0.           0.           0.

```

Comparando as matrizes Q encontradas, é possível ver que todas foram bem similares, sendo a maior diferença na ordem das matrizes, já que isso varia com base no método utilizado. Além disso, comparando o resultado obtido com a função *qr* nativa do *Scilab* com as demais, é possível notar que tanto seu Q quanto seu R são exatamente iguais aos seus respectivos obtidos pelo método de Householder. Dessa maneira, é possível elaborar a hipótese que a função nativa é feita através do método de Householder.

5 Algoritmo QR para Autovalores

Para a questão 5, foi desenvolvido uma função para calcular os autovalores de uma matriz simétrica A através do Algoritmo *QR*, que pode ser encontrada no arquivo *QR.para_Autovalores.sci*. Note que, para o funcionamento do método, a função $[Q, R] = qr_GS(A)$ do arquivo *Metodo_Gram_Schmidt.sci* também precisa ser executada.

```

1 --> A=[1 2 3; 1 2 2; 4 5 3]
2 A  =
3
4  1.    2.    3.
5  1.    2.    2.
6  4.    5.    3.
7
8 --> A';
9
10 --> A=A'*A
11 A  =
12
13  18.    24.    17.
14  24.    33.    25.
15  17.    25.    22.
16

```



```

17 --> [S] = espectro(A, 10^-3)
18 S =
19
20      69.762275
21      3.1973761
22      0.0403486
23
24 --> [S] = espectro(A, 10^-4)
25 S =
26
27      69.762276
28      3.1973753
29      0.0403486
30
31 --> [S] = espectro(A, 10^-5)
32 S =
33
34      69.762276
35      3.1973753
36      0.0403486
37 -----
38
39 --> A=[3 -2 4; -2 6 2; 4 2 3]
40 A =
41
42      3.   -2.   4.
43     -2.   6.   2.
44      4.   2.   3.
45
46 --> [S] = espectro(A, 10^-3)
47 S =
48
49      6.9999739
50      6.9999853
51     -1.9999592
52
53 --> [S] = espectro(A, 10^-4)
54 S =
55
56      6.9999979
57      6.9999988
58     -1.9999967
59
60 --> [S] = espectro(A, 10^-5)
61 S =
62
63      6.9999998
64      6.9999999
65     -1.9999997
66 -----
67
68 --> A=[1, 2, 3; 2, 4, 5; 3, 5, 6]
69 A =
70
71      1.   2.   3.
72      2.   4.   5.
73      3.   5.   6.

```

```

74 --> A';
75
76
77 --> [S] = espectro(A, 10^-3)
78 S =
79
80     11.344814
81     -0.5156938
82     0.1708796
83
84 --> [S] = espectro(A, 10^-4)
85 S =
86
87     11.344814
88     -0.5157256
89     0.1709113
90
91 --> [S] = espectro(A, 10^-5)
92 S =
93
94     11.344814
95     -0.515729
96     0.1709148

```

Observando os resultados acima e comparando com resultados previamente calculados, é possível ver que o método está entregando os resultados corretos. Inclusive, a última matriz usada, $A = [1, 2, 3; 2, 4, 5; 3, 5, 6]$, foi usada Na Aula Prática 3, com o intuito de obter os autovalores usando os Métodos da Potência Deslocada. Assim, comparando as duas respostas, é possível notar que estão praticamente as mesmas:

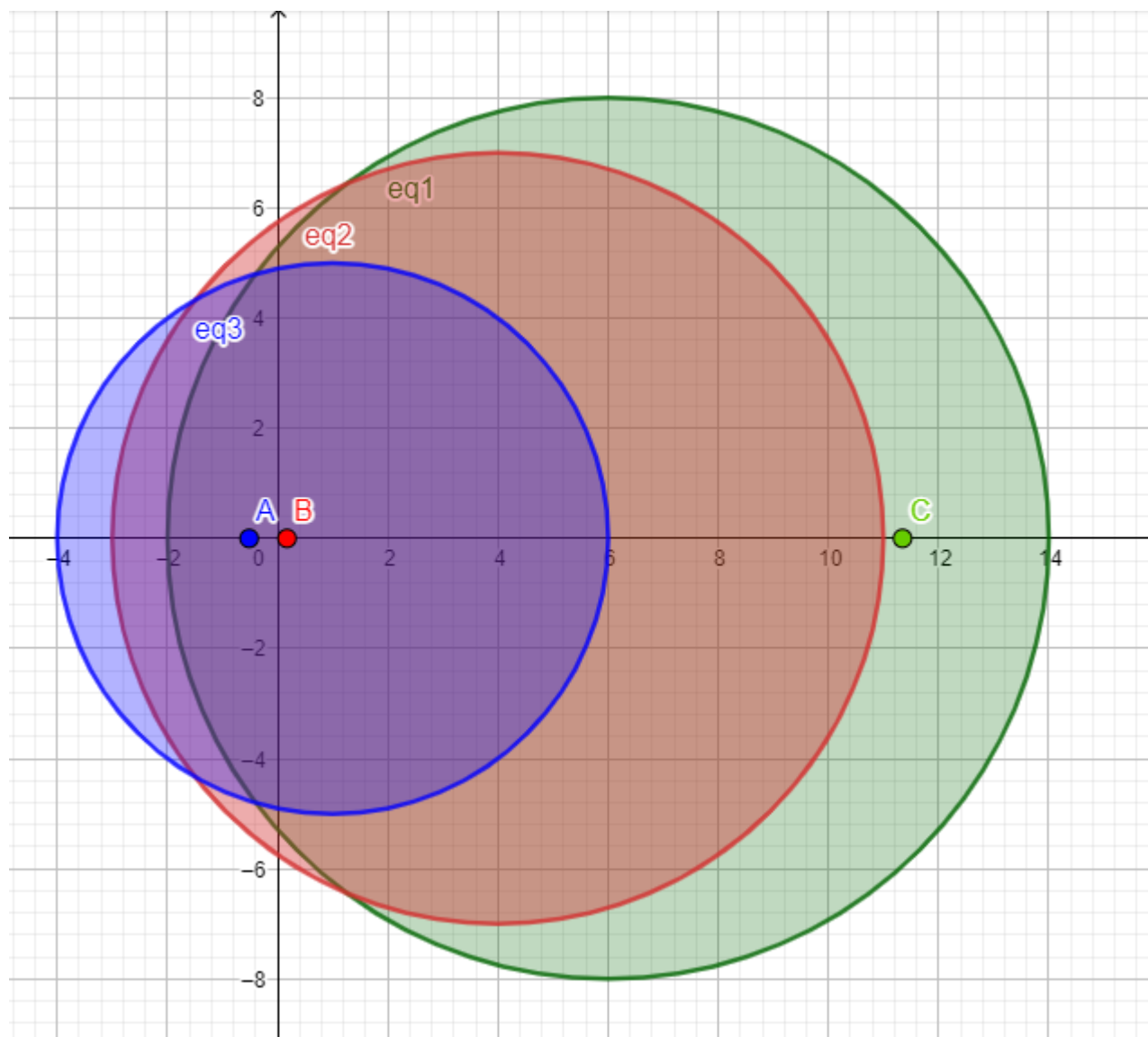
```

1 --> A=[1 2 3; 2 4 5; 3 5 6]
2 A =
3
4     1.     2.     3.
5     2.     4.     5.
6     3.     5.     6.
7
8 --> x0=[1;2;3]
9 x0 =
10
11     1.
12     2.
13     3.
14
15 --> [lambda]=Potencia_deslocada_inversa(A,x0,10^(-10),1,100)
16
17 "0 numero de iteracoes passou do maximo de iteracoes permitido"
18 lambda =
19
20     0.1709152
21
22 --> [lambda]=Potencia_deslocada_inversa(A,x0,10^(-10),6,100)
23
24 "0 numero de iteracoes passou do maximo de iteracoes permitido"
25 lambda =
26

```

```
27     11.344814
28
29 --> [lambda]=Potencia_deslocada_inversa(A,x0,10^(-10),-1,100)
30
31     "0 numero de iteracoes passou do maximo de iteracoes permitido"
32     lambda =
33
34     -0.5157295
```

Podemos ainda ver os Discos de Gerschgorin dessa matriz:



Discos de Gerschgorin calculados com os autovalores indicados

6 Testes extras

As matrizes usadas nessa aula prática não são tão grandes assim, têm uma ordem pequena demais! Que tal testarmos com umas matrizes com um n um pouco maior que 4?

```
1 --> A=rand(10,10);
```

```

2
3 --> [Q,R]=qr_GS(A);
4
5 --> Q';
6
7 --> QtQ=Q'*Q;
8
9 --> I=eye(10,10);
10
11 --> norm(QtQ-I)
12     ans =
13
14     1.198D-14
15
16 --> norm(A-Q*R)
17     ans =
18
19     2.771D-16
20 -----
21
22 --> [Q,R]=qr_GSM(A);
23
24 --> Q';
25
26 --> QtQ=Q'*Q;
27
28 --> I=eye(10,10);
29
30 --> norm(QtQ-I)
31     ans =
32
33     2.434D-15
34
35 --> norm(A-Q*R)
36     ans =
37
38     3.913D-16
39 -----
40
41 --> [U,R]=qr_House(A);
42
43 --> [Q]=constroi_Q_House(U);
44
45 --> Q';
46
47 --> QtQ=Q'*Q;
48
49 --> I=eye(10,10);
50
51 --> norm(QtQ-I)
52     ans =
53
54     1.579D-15
55
56 --> norm(A-Q*R)
57     ans =
58

```

```

59      0.7853395
60      -----
61      -----
62  --> A=rand(100,100);
63
64  --> [Q,R]=qr_GS(A);
65
66  --> Q';
67
68  --> QtQ=Q'*Q;
69
70  --> I=eye(100,100);
71
72  --> norm(QtQ-I)
73  ans =
74
75      3.262D-12
76
77  --> norm(A-Q*R)
78  ans =
79
80      4.298D-15
81  -----
82
83  --> [Q,R]=qr_GSM(A);
84
85  --> Q';
86
87  --> QtQ=Q'*Q;
88
89  --> I=eye(100,100);
90
91  --> norm(QtQ-I)
92  ans =
93
94      7.166D-14
95
96  --> norm(A-Q*R)
97  ans =
98
99      4.255D-15
100 -----
101
102 --> [U,R]=qr_House(A);
103
104 --> [Q]=constroi_Q_House(U);
105
106 --> Q';
107
108 --> QtQ=Q'*Q;
109
110 --> I=eye(100,100);
111
112 --> norm(QtQ-I)
113 ans =
114
115      6.960D-15

```

```

116
117 --> norm(A-Q*R)
118 ans =
119
120 0.2385835
121 -----
122 -----
123 --> A=rand(1000,1000);
124
125 --> [Q,R]=qr_GS(A);
126
127 --> Q';
128
129 --> QtQ=Q'*Q;
130
131 --> I=eye(1000,1000);
132
133 --> norm(QtQ-I)
134 ans =
135
136 5.525D-11
137
138 --> norm(A-Q*R)
139 ans =
140
141 2.663D-14
142 -----
143
144 --> [Q,R]=qr_GSM(A);
145
146 --> Q';
147
148 --> QtQ=Q'*Q;
149
150 --> I=eye(1000,1000);
151
152 --> norm(QtQ-I)
153 ans =
154
155 4.314D-13
156
157 --> norm(A-Q*R)
158 ans =
159
160 2.689D-14
161 -----
162
163 --> [U,R]=qr_House(A);
164
165 --> [Q]=constroi_Q_House(U);
166
167 --> Q';
168
169 --> QtQ=Q'*Q;
170
171 --> I=eye(1000,1000);
172

```

```

173 --> norm(QtQ-I)
174 ans =
175
176     5.149D-14
177
178 --> norm(A-Q*R)
179 ans =
180
181     0.9661908

```

É, pelo visto os métodos estão bem adequados sim! Os maiores valores encontrados foram na norma de $A - Q * R$ calculadas pelo método de Householder, mas ainda assim, os valores estão próximos de 1 no máximo.

Para achar os autovalores de matrizes aleatórias simétricas, eu irei usar matrizes um pouco menores, para que não precisem passar por 100 ou 1000 autovalores.

```

1 --> A=rand(5,5);
2
3 --> A';
4
5 --> A=A*A';
6
7 --> [S] = espectro(A, 10^-3)
8 S =
9
10     5.2669802
11     1.0264177
12     0.3283297
13     0.0415092
14     0.0011334
15
16 --> [S] = espectro(A, 10^-4)
17 S =
18
19     5.2670008
20     1.0264014
21     0.3283258
22     0.0415089
23     0.0011334
24
25 --> [S] = espectro(A, 10^-5)
26 S =
27
28     5.2670015
29     1.0264011
30     0.3283254
31     0.0415089
32     0.0011334
33 -----
34
35 --> A=rand(10,10);
36
37 --> A';
38
39 --> A=A*A';
40

```



```

41 --> [S] = espectro(A, 10^-3)
42 S =
43
44     25.892728
45     1.9630866
46     1.4684304
47     0.8863556
48     0.7919037
49     0.5540814
50     0.4200061
51     0.1062325
52     0.0275176
53     0.0105287
54
55 --> [S] = espectro(A, 10^-4)
56 S =
57
58     25.892728
59     1.963803
60     1.467714
61     0.8880484
62     0.7902123
63     0.5541254
64     0.4199607
65     0.1062325
66     0.0275176
67     0.0105287
68
69 --> [S] = espectro(A, 10^-5)
70 S =
71
72     25.892728
73     1.9638098
74     1.4677072
75     0.8883221
76     0.7899386
77     0.554126
78     0.4199601
79     0.1062325
80     0.0275176
81     0.0105287
82 -----
83
84 --> A=rand(15,15);
85
86 --> A';
87
88 --> A=A*A';
89
90 --> [S] = espectro(A, 10^-3)
91 S =
92
93     57.425252
94     4.6975307
95     2.6063016
96     2.4949439
97     1.7115712

```

```

98      1.4843662
99      1.1075509
100     0.8997718
101     0.5878988
102     0.4654406
103     0.2729977
104     0.1575734
105     0.0757784
106     0.0328802
107     0.0085217
108
109 --> [S] = espectro(A, 10^-4)
110 S =
111
112     57.425252
113     4.6975307
114     2.6152129
115     2.4860326
116     1.7115712
117     1.4843662
118     1.1075509
119     0.8997718
120     0.5878988
121     0.4654406
122     0.2729977
123     0.1575734
124     0.0757784
125     0.0328802
126     0.0085217
127
128 --> [S] = espectro(A, 10^-5)
129 S =
130
131     57.425252
132     4.6975307
133     2.6160228
134     2.4852227
135     1.7115712
136     1.4843662
137     1.1075509
138     0.8997718
139     0.5878988
140     0.4654406
141     0.2729977
142     0.1575734
143     0.0757784
144     0.0328802
145     0.0085217
146 -----
147
148 --> A=rand(20,20);
149
150 --> A';
151
152 --> A=A*A';
153
154 --> [S] = espectro(A, 10^-3)

```

```

155 S =
156
157 93.97626
158 5.5219113
159 3.7933611
160 3.0702844
161 2.7516485
162 2.3993211
163 1.9716616
164 1.6524285
165 1.5105768
166 1.4382597
167 1.1433269
168 0.8530193
169 0.6748887
170 0.6022279
171 0.4001912
172 0.2324238
173 0.1520201
174 0.024632
175 0.0137778
176 0.0005522
177
178 --> [S] = espectro(A, 10^-4)
179 S =
180
181 93.97626
182 5.5219113
183 3.7933613
184 3.0702882
185 2.7516471
186 2.3993184
187 1.9716624
188 1.6568548
189 1.5061766
190 1.4382329
191 1.1433269
192 0.8530193
193 0.6748888
194 0.6022278
195 0.4001912
196 0.2324238
197 0.1520201
198 0.024632
199 0.0137778
200 0.0005522
201
202 --> [S] = espectro(A, 10^-5)
203 S =
204
205 93.97626
206 5.5219113
207 3.7933613
208 3.0702884
209 2.751647
210 2.3993184
211 1.9716624

```

212	1.6572253
213	1.5058138
214	1.4382251
215	1.1433269
216	0.8530193
217	0.6748888
218	0.6022277
219	0.4001912
220	0.2324238
221	0.1520201
222	0.024632
223	0.0137778
224	0.0005522

Fontes de Consulta

- Aulas Gravadas
- Teste 2 (Usei as matrizes Q e R do teste como primeiro exemplo)
- The QR Decomposition (<https://www.youtube.com/watch?v=1xcSttdeHFg>)
- QR decomposition of a 4x4 Matrix with the Householder Transformation (https://www.youtube.com/watch?v=s_rhzm6uW_E)
- QR Factorization (<https://www.youtube.com/watch?v=qmRC8mTPGI8>)
- QR decomposition (https://en.wikipedia.org/wiki/QR_decomposition)
- Matrizes simétricas (https://www.ufrgs.br/reatmat/AlgebraLinear/livro/s1e1-matrizes_simx00e9tricas.html)