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*.

```
--> Q=[-0.8 0.546; 0.6 0.728; 0 -0.414]
 Q  =

  -0.8   0.546
   0.6   0.728
   0.   -0.414

--> R=[5 0.4; 0 2.417]
 R  =

   5.   0.4
   0.   2.417

--> A=Q*R
 A  =

  -4.   0.999682
   3.   1.999576
   0.  -1.000638

--> [Q,R]=qr_GS(A)
 Q  =

  -0.8   0.5461376
   0.6   0.7281835
   0.   -0.4141044
```

```
 R  =

   5.   0.4
   0.   2.4163908

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          5.551D-17
   5.551D-17   1.

--> I=eye(2,2)
 I  =

   1.   0.
   0.   1.

--> norm(QtQ-I)
 ans  =

   5.551D-17

--> norm(A-Q*R)
 ans  =

   0.
--------------------------------------------------

--> A=[4 9 -14; 3 13 2; 0 5 0]
 A  =

   4.   9.   -14.
   3.   13.   2.
   0.   5.    0.

--> [Q,R]=qr_GS(A)
 Q  =

   0.8  -0.4242641  -0.4242641
   0.6   0.5656854   0.5656854
   0.    0.7071068  -0.7071068
 R  =

   5.   15.        -10.
   0.   7.0710678   7.0710678
   0.   0.          7.0710678

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.         -5.551D-17   2.220D-16
  -5.551D-17   1.          1.015D-17
   1.984D-16  -4.536D-17   1.
```

```

--> I=eye(3,3)
 I  =

   1.   0.   0.
   0.   1.   0.
   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   2.331D-16

--> norm(A-Q*R)
 ans  =

   4.555D-16
--------------------------------------------------

--> A=[1 0 1 1; 0 1 1 1; 1 1 1 0; 1 1 0 1]
 A  =

   1.   0.   1.   1.
   0.   1.   1.   1.
   1.   1.   1.   0.
   1.   1.   0.   1.

--> [Q,R]=qr_GS(A)
 Q  =

   0.5773503  -0.5163978   0.5070926   0.3779645
   0.          0.7745967   0.5070926   0.3779645
   0.5773503   0.2581989   0.1690309  -0.7559289
   0.5773503   0.2581989  -0.6761234   0.3779645
 R  =

   1.7320508   1.1547005   1.1547005   1.1547005
   0.          1.2909944   0.5163978   0.5163978
   0.          0.          1.183216    0.3380617
   0.          0.          0.          1.1338934

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          -2.321D-16  -2.128D-16  -1.951D-16
  -2.321D-16   1.           1.899D-16   1.645D-16
  -2.128D-16   1.899D-16   1.          -2.077D-17
  -1.951D-16   1.645D-16  -2.077D-17   1.

--> I=eye(4,4)
 I  =

   1.   0.   0.   0.
   0.   1.   0.   0.
   0.   0.   1.   0.
```

```
   0.   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   6.003D-16

--> norm(A-Q*R)
 ans  =

   1.118D-16
--------------------------------------------------

--> A=[1 2 3; -1 1 1; 1 1 1; 1 1 1]
 A  =

   1.   2.   3.
  -1.   1.   1.
   1.   1.   1.
   1.   1.   1.

--> [Q,R]=qr_GS(A)
 Q  =

   0.5   0.5735393   0.6488857
  -0.5   0.8029551  -0.3244428
   0.5   0.1147079  -0.4866643
   0.5   0.1147079  -0.4866643
 R  =

   2.   1.5         2.
   0.   2.1794495   2.7529888
   0.   0.          0.6488857

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.   0.          0.
   0.   1.         -1.277D-15
   0.  -1.265D-15   1.

--> I=eye(3,3)
 I  =

   1.   0.   0.
   0.   1.   0.
   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   1.512D-15

--> norm(A-Q*R)
 ans  =
```

```
   0.
```

Observando os testes feitos, incluindo a ortogonalidade das matrizes Q^TQ calculadas, podemos ver que o método é de fato preciso, já que Q^TQ 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^TQ-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 $Metodo\_Gram\_Schmidt\_Mod.sci$.

```
--> Q=[-0.8 0.546; 0.6 0.728; 0 -0.414]
 Q  =

  -0.8   0.546
   0.6   0.728
   0.   -0.414

--> R=[5 0.4; 0 2.417]
 R  =

   5.   0.4
   0.   2.417

--> A=Q*R
 A  =

  -4.   0.999682
   3.   1.999576
   0.  -1.000638

--> [Q,R]=qr_GSM(A)
 Q  =

  -0.8   0.5461376
   0.6   0.7281835
   0.   -0.4141044
 R  =

   5.   0.4
   0.   2.4163908

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          5.551D-17
   5.551D-17   1.
```

```

--> I=eye(2,2)
 I  =

   1.   0.
   0.   1.

--> norm(QtQ-I)
 ans  =

   5.551D-17

--> norm(A-Q*R)
 ans  =

   0.
-----------------------------------------------------

--> A=[4 9 -14; 3 13 2; 0 5 0]
 A  =

   4.   9.   -14.
   3.   13.   2.
   0.   5.    0.

--> [Q,R]=qr_GSM(A)
 Q  =

   0.8  -0.4242641  -0.4242641
   0.6   0.5656854   0.5656854
   0.    0.7071068  -0.7071068
 R  =

   5.   15.         -10.
   0.   7.0710678   7.0710678
   0.   0.          7.0710678

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          -5.551D-17   1.665D-16
  -5.551D-17   1.           8.865D-17
   1.429D-16   8.865D-17   1.

--> I=eye(3,3)
 I  =

   1.   0.   0.
   0.   1.   0.
   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   2.838D-16
```

```

--> norm(A-Q*R)
 ans  =

   8.882D-16
--------------------------------------------------

--> A=[1 0 1 1; 0 1 1 1; 1 1 1 0; 1 1 0 1]
 A  =

   1.   0.   1.   1.
   0.   1.   1.   1.
   1.   1.   1.   0.
   1.   1.   0.   1.

--> [Q,R]=qr_GSM(A)
 Q  =

   0.5773503  -0.5163978   0.5070926   0.3779645
   0.          0.7745967   0.5070926   0.3779645
   0.5773503   0.2581989   0.1690309  -0.7559289
   0.5773503   0.2581989  -0.6761234   0.3779645
 R  =

   1.7320508   1.1547005   1.1547005   1.1547005
   0.          1.2909944   0.5163978   0.5163978
   0.          0.          1.183216    0.3380617
   0.          0.          0.          1.1338934

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.         -2.321D-16  -2.214D-16  -1.759D-16
  -2.321D-16   1.         -8.856D-17  -7.230D-17
  -2.214D-16  -8.856D-17   1.          6.784D-17
  -1.759D-16  -7.230D-17   6.784D-17   1.

--> I=eye(4,4)
 I  =

   1.   0.   0.   0.
   0.   1.   0.   0.
   0.   0.   1.   0.
   0.   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   4.563D-16

--> norm(A-Q*R)
 ans  =

   1.578D-16
--------------------------------------------------
```

```
--> A=[1 2 3; -1 1 1; 1 1 1; 1 1 1]
 A  =

   1.   2.   3.
  -1.   1.   1.
   1.   1.   1.
   1.   1.   1.

--> [Q,R]=qr_GSM(A)
 Q  =

   0.5   0.5735393   0.6488857
  -0.5   0.8029551  -0.3244428
   0.5   0.1147079  -0.4866643
   0.5   0.1147079  -0.4866643
 R  =

   2.   1.5         2.
   0.   2.1794495   2.7529888
   0.   0.          0.6488857

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.   0.          0.
   0.   1.         -1.277D-15
   0.  -1.265D-15   1.

--> I=eye(3,3)
 I  =

   1.   0.   0.
   0.   1.   0.
   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   1.512D-15

--> norm(A-Q*R)
 ans  =

   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 $Metodo\_Householder.sci$.

```
--> A=[-4. 0.999682; 3. 1.999576; 0. -1.000638]
 A  =

  -4.    0.999682
   3.    1.999576
   0.   -1.000638

--> [U,R]=qr_House(A)
 U  =

  -0.9486833   0.
   0.3162278   0.9772997
   0.         -0.2118615
 R  =

   5.    0.4
   0.   -2.4163908
   0.    0.

--> [Q]=constroi_Q_House(U)
 Q  =

  -0.8  -0.5461376   0.2484626
   0.6  -0.7281835   0.3312835
   0.    0.4141044   0.9102294

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.         -1.665D-16   5.551D-17
  -1.665D-16   1.         -2.601D-17
   6.481D-17  -8.152D-17   1.

--> I=eye(3,3)
 I  =

   1.    0.    0.
   0.    1.    0.
```

```
   0.   0.   1.

--> norm(QtQ-I)
 ans  =

   3.843D-16

--> norm(A-Q*R)
 ans  =

   1.099D-15
--------------------------------------------------

--> A=[4 9 -14; 3 13 2; 0 5 0]
 A  =

   4.   9.   -14.
   3.   13.   2.
   0.   5.    0.

--> [U,R]=qr_House(A)
 U  =

   0.9486833   0.          0.
   0.3162278   0.9238795   0.
   0.          0.3826834   1.
 R  =

  -5.  -15.          10.
   0.  -7.0710678  -7.0710678
   0.   0.         -7.0710678

--> [Q]=constroi_Q_House(U)
 Q  =

  -0.8   0.4242641  -0.4242641
  -0.6  -0.5656854   0.5656854
   0.   -0.7071068  -0.7071068

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          1.110D-16  -1.665D-16
   1.110D-16   1.         -1.015D-17
  -1.429D-16  -1.015D-17   1.

--> I=eye(3,3)
 I  =

   1.   0.   0.
   0.   1.   0.
   0.   0.   1.

--> norm(QtQ-I)
 ans  =
```

```

   4.183D-16

--> norm(A-Q*R)
 ans  =

   14.142136
--------------------------------------------------

--> A=[1 0 1 1; 0 1 1 1; 1 1 1 0; 1 1 0 1]
 A  =

   1.   0.   1.   1.
   0.   1.   1.   1.
   1.   1.   1.   0.
   1.   1.   0.   1.

--> [U,R]=qr_House(A)
 U  =

   0.8880738   0.           0.           0.
   0.          0.9419651   0.           0.
   0.3250576   0.2373833  -0.756429     0.
   0.3250576   0.2373833  -0.6540758   1.
 R  =

  -1.7320508  -1.1547005  -1.1547005  -1.1547005
   0.         -1.2909944  -0.5163978  -0.5163978
   0.          0.          1.183216    0.3380617
   0.          0.          0.          1.1338934

--> [Q]=constroi_Q_House(U)
 Q  =

  -0.5773503   0.5163978   0.5070926  -0.3779645
   0.         -0.7745967   0.5070926  -0.3779645
  -0.5773503  -0.2581989   0.1690309   0.7559289
  -0.5773503  -0.2581989  -0.6761234  -0.3779645

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          3.235D-16   2.761D-16  -2.000D-16
   3.235D-16   1.         -5.110D-16   4.081D-16
   2.761D-16  -5.110D-16   1.          2.188D-16
  -2.000D-16   4.081D-16   2.188D-16   1.

--> I=eye(4,4)
 I  =

   1.   0.   0.   0.
   0.   1.   0.   0.
   0.   0.   1.   0.
   0.   0.   0.   1.

```

```
--> norm(QtQ-I)
 ans  =

   1.758D-15

--> norm(A-U*R)
 ans  =

   6.2139499
---------------------------------------------------

--> A=[1 2 3; -1 1 1; 1 1 1; 1 1 1]
 A  =

   1.   2.   3.
  -1.   1.   1.
   1.   1.   1.
   1.   1.   1.

--> [U,R]=qr_House(A)
 U  =

   0.8660254   0.          0.
  -0.2886751   0.9985326   0.
   0.2886751  -0.0382921  -0.9238795
   0.2886751  -0.0382921  -0.3826834
 R  =

  -2.  -1.5        -2.
   0.  -2.1794495  -2.7529888
   0.   0.          0.6488857
   0.   0.          0.

--> [Q]=constroi_Q_House(U)
 Q  =

  -0.5  -0.5735393   0.6488857  -3.514D-17
   0.5  -0.8029551  -0.3244428   3.522D-17
  -0.5  -0.1147079  -0.4866643  -0.7071068
  -0.5  -0.1147079  -0.4866643   0.7071068

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.         -5.667D-16   6.413D-16  -9.524D-18
  -5.667D-16   1.          2.292D-16  -1.438D-17
   6.413D-16   2.292D-16   1.         -1.179D-16
  -9.524D-18  -1.438D-17  -1.179D-16   1.

--> I=eye(4,4)
 I  =

   1.   0.   0.   0.
   0.   1.   0.   0.
   0.   0.   1.   0.
```

```
   0.    0.    0.    1.

--> norm(QtQ-I)
 ans  =

   1.739D-15

--> norm(A-Q*R)
 ans  =

   6.639D-15
```

4.1

Comparando as matrizes $Q^T Qs$ obtidas diretamente das matrizes A pelo método de Gram-Schmidt e as $Q^T Qs$ 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 $Q^T Qs$. Pelo método de Gram-Schmidt, se A é uma matriz mxn, $Q^T Qs$ será nxn, 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 mxm, consequentemente Q também o será.

4.2

```
--> A = [0.70000 0.70711; 0.70001 0.70711]
 A  =

   0.7        0.70711
   0.70001    0.70711

--> [Q,R]=qr_GS(A)
 Q  =

   0.7071017   0.7071118
   0.7071118  -0.7071017
 R  =

   0.9899566   1.0000046
   0.          0.0000071

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          2.301D-11
   2.301D-11   1.
--------------------------------------------------

```

```
--> [Q,R]=qr_GSM(A)
 Q  =

   0.7071017   0.7071118
   0.7071118  -0.7071017
 R  =

   0.9899566   1.0000046
   0.          0.0000071

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.          2.301D-11
   2.301D-11   1.
--------------------------------------------------

--> [U,R]=qr_House(A)
 U  =

   0.9238782   0.
   0.3826867   1.
 R  =

  -0.9899566  -1.0000046
   0.         -0.0000071

--> [Q]=constroi_Q_House(U)
 Q  =

  -0.7071017   0.7071118
  -0.7071118  -0.7071017

--> Q';

--> QtQ=Q'*Q
 QtQ  =

   1.   0.
   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^TQ 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^TQ com Q obtido pela matriz U foi exatamente uma matriz identidade.

4.3

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

```
  1.   2.   3.
  4.   5.   6.
  7.   8.   7.
  4.   2.   3.
  4.   2.   2.

--> [Q,R]=qr_GS(A)
 Q  =

  0.1010153   0.3161731   0.5419969
  0.404061    0.3533699   0.5161875
  0.7071068   0.3905667  -0.5247906
  0.404061   -0.5579525   0.3871406
  0.404061   -0.5579525  -0.1204438
 R  =

  9.8994949   9.4954339   9.6974644
  0.          3.2919196   3.0129434
  0.          0.          1.9701157

--> [Q,R]=qr_GSM(A)
 Q  =

  0.1010153   0.3161731   0.5419969
  0.404061    0.3533699   0.5161875
  0.7071068   0.3905667  -0.5247906
  0.404061   -0.5579525   0.3871406
  0.404061   -0.5579525  -0.1204438
 R  =

  9.8994949   9.4954339   9.6974644
  0.          3.2919196   3.0129434
  0.          0.          1.9701157

--> [U,R]=qr_House(A)
 U  =

  0.741962    0.          0.
  0.2722923   0.7865551   0.
  0.4765114   0.1191972  -0.9800408
  0.2722923  -0.4284409   0.1842055
  0.2722923  -0.4284409  -0.0747553
 R  =

 -9.8994949  -9.4954339  -9.6974644
  0.         -3.2919196  -3.0129434
  0.          0.          1.9701157
  0.          0.          0.
  0.          0.          0.

--> [Q]=constroi_Q_House(U)
 Q  =

 -0.1010153  -0.3161731   0.5419969  -0.6842085  -0.3576711
 -0.404061   -0.3533699   0.5161875   0.3280084   0.5812274
 -0.7071068  -0.3905667  -0.5247906   0.0093972  -0.2682612
```

```
  -0.404061    0.5579525   0.3871406   0.3655973  -0.4918175
  -0.404061    0.5579525  -0.1204438  -0.5389987   0.4694651

--> [Q,R]=qr(A)
 Q  =

  -0.1010153  -0.3161731   0.5419969  -0.6842085  -0.3576711
  -0.404061   -0.3533699   0.5161875   0.3280084   0.5812274
  -0.7071068  -0.3905667  -0.5247906   0.0093972  -0.2682612
  -0.404061    0.5579525   0.3871406   0.3655973  -0.4918175
  -0.404061    0.5579525  -0.1204438  -0.5389987   0.4694651
 R  =

  -9.8994949  -9.4954339  -9.6974644
   0.         -3.2919196  -3.0129434
   0.          0.          1.9701157
   0.          0.          0.
   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.

```
--> A=[1 2 3; 1 2 2; 4 5 3]
 A  =

   1.   2.   3.
   1.   2.   2.
   4.   5.   3.

--> A';

--> A=A'*A
 A  =

   18.   24.   17.
   24.   33.   25.
   17.   25.   22.

```

```
--> [S] = espectro(A, 10^-3)
 S  =

   69.762275
   3.1973761
   0.0403486

--> [S] = espectro(A, 10^-4)
 S  =

   69.762276
   3.1973753
   0.0403486

--> [S] = espectro(A, 10^-5)
 S  =

   69.762276
   3.1973753
   0.0403486
--------------------------------------------------

--> A=[3 -2 4; -2 6 2; 4 2 3]
 A  =

   3.  -2.   4.
  -2.   6.   2.
   4.   2.   3.

--> [S] = espectro(A, 10^-3)
 S  =

   6.9999739
   6.9999853
  -1.9999592

--> [S] = espectro(A, 10^-4)
 S  =

   6.9999979
   6.9999988
  -1.9999967

--> [S] = espectro(A, 10^-5)
 S  =

   6.9999998
   6.9999999
  -1.9999997
--------------------------------------------------

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

   1.   2.   3.
   2.   4.   5.
   3.   5.   6.
```

```

--> A';

--> [S] = espectro(A, 10^-3)
 S  =

   11.344814
  -0.5156938
   0.1708796

--> [S] = espectro(A, 10^-4)
 S  =

   11.344814
  -0.5157256
   0.1709113

--> [S] = espectro(A, 10^-5)
 S  =

   11.344814
  -0.515729
   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:

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

   1.   2.   3.
   2.   4.   5.
   3.   5.   6.

--> x0=[1;2;3]
 x0  =

   1.
   2.
   3.

--> [lambda]=Potencia_deslocada_inversa(A,x0,10^(-10),1,100)

  "O numero de iteracoes passou do maximo de iteracoes permitido"
 lambda  =

   0.1709152

--> [lambda]=Potencia_deslocada_inversa(A,x0,10^(-10),6,100)

  "O numero de iteracoes passou do maximo de iteracoes permitido"
 lambda  =

```

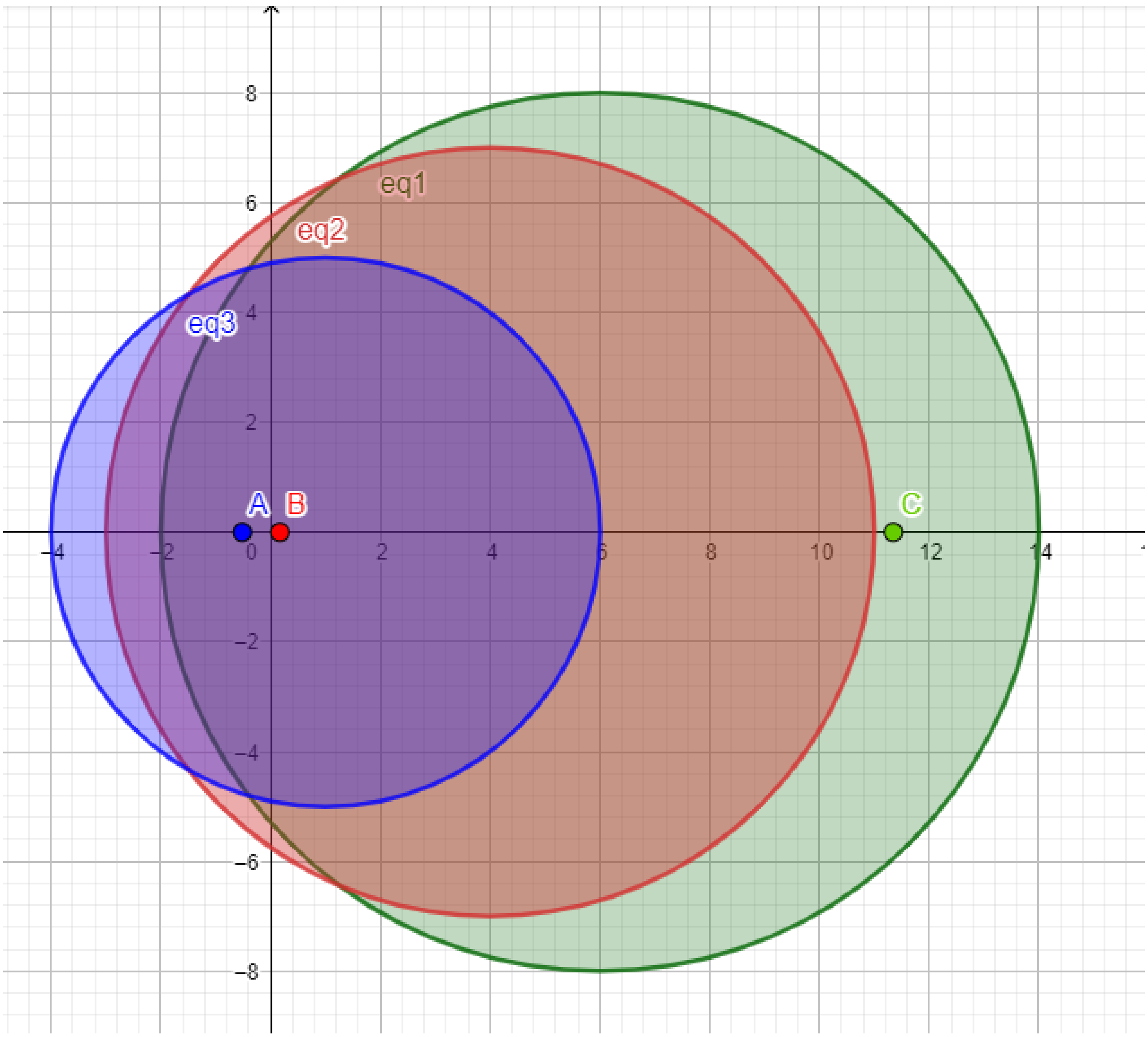
```
   11.344814

--> [lambda]=Potencia_deslocada_inversa(A,x0,10^(-10),-1,100)

  "O numero de iteracoes passou do maximo de iteracoes permitido"
 lambda  =

  -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?

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

```

--> [Q,R]=qr_GS(A);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(10,10);

--> norm(QtQ-I)
 ans  =

   1.198D-14

--> norm(A-Q*R)
 ans  =

   2.771D-16
--------------------------------------------------

--> [Q,R]=qr_GSM(A);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(10,10);

--> norm(QtQ-I)
 ans  =

   2.434D-15

--> norm(A-Q*R)
 ans  =

   3.913D-16
--------------------------------------------------

--> [U,R]=qr_House(A);

--> [Q]=constroi_Q_House(U);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(10,10);

--> norm(QtQ-I)
 ans  =

   1.579D-15

--> norm(A-Q*R)
 ans  =

```

```
   0.7853395
---------------------------------------------------
---------------------------------------------------
--> A=rand(100,100);

--> [Q,R]=qr_GS(A);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(100,100);

--> norm(QtQ-I)
 ans  =

   3.262D-12

--> norm(A-Q*R)
 ans  =

   4.298D-15
---------------------------------------------------

--> [Q,R]=qr_GSM(A);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(100,100);

--> norm(QtQ-I)
 ans  =

   7.166D-14

--> norm(A-Q*R)
 ans  =

   4.255D-15
---------------------------------------------------

--> [U,R]=qr_House(A);

--> [Q]=constroi_Q_House(U);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(100,100);

--> norm(QtQ-I)
 ans  =

   6.960D-15
```

```

--> norm(A-Q*R)
 ans  =

   0.2385835
--------------------------------------------------
--------------------------------------------------
--> A=rand(1000,1000);

--> [Q,R]=qr_GS(A);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(1000,1000);

--> norm(QtQ-I)
 ans  =

   5.525D-11

--> norm(A-Q*R)
 ans  =

   2.663D-14
--------------------------------------------------

--> [Q,R]=qr_GSM(A);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(1000,1000);

--> norm(QtQ-I)
 ans  =

   4.314D-13

--> norm(A-Q*R)
 ans  =

   2.689D-14
--------------------------------------------------

--> [U,R]=qr_House(A);

--> [Q]=constroi_Q_House(U);

--> Q';

--> QtQ=Q'*Q;

--> I=eye(1000,1000);

```

```
--> norm(QtQ-I)
 ans  =

   5.149D-14

--> norm(A-Q*R)
 ans  =

   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.

```
--> A=rand(5,5);

--> A';

--> A=A*A';

--> [S] = espectro(A, 10^-3)
 S  =

   5.2669802
   1.0264177
   0.3283297
   0.0415092
   0.0011334

--> [S] = espectro(A, 10^-4)
 S  =

   5.2670008
   1.0264014
   0.3283258
   0.0415089
   0.0011334

--> [S] = espectro(A, 10^-5)
 S  =

   5.2670015
   1.0264011
   0.3283254
   0.0415089
   0.0011334
--------------------------------------------------

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

--> A';

--> A=A*A';

```

```
--> [S] = espectro(A, 10^-3)
 S  =

   25.892728
   1.9630866
   1.4684304
   0.8863556
   0.7919037
   0.5540814
   0.4200061
   0.1062325
   0.0275176
   0.0105287

--> [S] = espectro(A, 10^-4)
 S  =

   25.892728
   1.963803
   1.467714
   0.8880484
   0.7902123
   0.5541254
   0.4199607
   0.1062325
   0.0275176
   0.0105287

--> [S] = espectro(A, 10^-5)
 S  =

   25.892728
   1.9638098
   1.4677072
   0.8883221
   0.7899386
   0.554126
   0.4199601
   0.1062325
   0.0275176
   0.0105287
--------------------------------------------------

--> A=rand(15,15);

--> A';

--> A=A*A';

--> [S] = espectro(A, 10^-3)
 S  =

   57.425252
   4.6975307
   2.6063016
   2.4949439
   1.7115712
```

```
   1.4843662
   1.1075509
   0.8997718
   0.5878988
   0.4654406
   0.2729977
   0.1575734
   0.0757784
   0.0328802
   0.0085217

--> [S] = espectro(A, 10^-4)
 S  =

   57.425252
   4.6975307
   2.6152129
   2.4860326
   1.7115712
   1.4843662
   1.1075509
   0.8997718
   0.5878988
   0.4654406
   0.2729977
   0.1575734
   0.0757784
   0.0328802
   0.0085217

--> [S] = espectro(A, 10^-5)
 S  =

   57.425252
   4.6975307
   2.6160228
   2.4852227
   1.7115712
   1.4843662
   1.1075509
   0.8997718
   0.5878988
   0.4654406
   0.2729977
   0.1575734
   0.0757784
   0.0328802
   0.0085217
-------------------------------------------------

--> A=rand(20,20);

--> A';

--> A=A*A';

--> [S] = espectro(A, 10^-3)
```

```
 S  =

   93.97626
   5.5219113
   3.7933611
   3.0702844
   2.7516485
   2.3993211
   1.9716616
   1.6524285
   1.5105768
   1.4382597
   1.1433269
   0.8530193
   0.6748887
   0.6022279
   0.4001912
   0.2324238
   0.1520201
   0.024632
   0.0137778
   0.0005522

--> [S] = espectro(A, 10^-4)
 S  =

   93.97626
   5.5219113
   3.7933613
   3.0702882
   2.7516471
   2.3993184
   1.9716624
   1.6568548
   1.5061766
   1.4382329
   1.1433269
   0.8530193
   0.6748888
   0.6022278
   0.4001912
   0.2324238
   0.1520201
   0.024632
   0.0137778
   0.0005522

--> [S] = espectro(A, 10^-5)
 S  =

   93.97626
   5.5219113
   3.7933613
   3.0702884
   2.751647
   2.3993184
   1.9716624
```

```
    1.6572253
    1.5058138
    1.4382251
    1.1433269
    0.8530193
    0.6748888
    0.6022277
    0.4001912
    0.2324238
    0.1520201
    0.024632
    0.0137778
    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/reamat/AlgebraLinear/livro/s1e1-matrizes_simx00e9tricas.html