Aula Prática 3 (ALN)

Raphael F. Levy

April 29, 2021

1 Introdução

Para a terceira Aula Prática, foram trabalhados diferentes métodos de potência, que se aplicam a matrizes nxn que tenham um autovalor dominante λ_1, e o método age iterativamente para produzir uma sequência de escalares que converge para o autovalor dominante, assim como uma sequência de vetores que convergem para o vetor associado ao autovalor dominante, chamado de autovetor dominante (v_1). Para simplificar, consideramos que, para que o método funcione de forma devida, a matriz A utilizada deve ser diagonalizável ($D = P^{-1}AP$).

Os métodos foram feitos usando algoritmos "padrões" e métodos deslocados, um com iteração inversa e outro usando a iteração de Rayleigh, sendo que os últimos usam respectivamente um valor para achar o autovalor de A mais próximo sem calcular a matriz inversa, enquanto o outro usa o quociente de Rayleigh: $\frac{{x_k}^T * A * x_k}{{x_k}^T * x_k} = \frac{{x_k+1}^T * x_k}{{x_k}^T * x_k}$

2 Testando o método da Potência (Questão 1)

Ambas as funções criadas para determinar o autovalor dominante da matriz A através do Método da Potência (Versão 1 e 2) estão no arquivo "Metodo_da_Potencia.sci".

Testando com diferentes matrizes e vetores iniciais teremos:

```
--> A=[3 1; 1 3] (Autovalores: 4,2)
 A  =

   3.   1.
   1.   3.

--> x0=[14;30]
 x0  =

   14.
   30.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

```

```
 "O metodo convergiu com a precisao passada"
 lambda  =

   4.0000000
 x1  =

   1.
   1.
 k  =

   34.
 n_erro  =

   8.467D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   4.
 x1  =

   0.7071068
   0.7071068
 k  =

   32.
 n_erro  =

   8.467D-11
-----------------------------------------------------------------

--> A=[5 2; 2 5] (Autovalores: 7,3)
 A  =

   5.   2.
   2.   5.

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

   1.
   3.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   7.0000000
 x1  =

   1.
   1.
 k  =

```

```
   29.
 n_erro  =

   6.631D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   7.
 x1  =

   0.7071068
   0.7071068
 k  =

   27.
 n_erro  =

   7.736D-11

-------------------------------------------------------------------

--> A=[-1 3 -1; -3 5 -1; -3 3 1] (Autovalores: 2, 2, 1)
 A  =

  -1.   3.  -1.
  -3.   5.  -1.
  -3.   3.   1.

--> x0=[1;3;5]
 x0  =

   1.
   3.
   5.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   2.0000000
 x1  =

   0.3333333
   0.6666667
   1.
 k  =

   33.
 n_erro  =

   5.174D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)
```

```
 "O metodo convergiu com a precisao passada"
 lambda  =

   2.0000000
 x1  =

   0.2672612
   0.5345225
   0.8017837
 k  =

   30.
 n_erro  =

   8.147D-11

-------------------------------------------------------------------

--> A=[1 2 0; 0 3 0; 2 -4 2] (Autovalores: 3, 2, 1)
 A  =

   1.    2.    0.
   0.    3.    0.
   2.   -4.    2.

--> x0=[3;4;5]
 x0  =

   3.
   4.
   5.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

  -0.5
  -0.5
   1.
 k  =

   57.
 n_erro  =

   7.099D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
```

```
 x1  =

   0.4082483
   0.4082483
  -0.8164966
 k  =

   55.
 n_erro  =

   6.693D-11

-------------------------------------------------------------------

--> A=[1 0 2 0; 2 1 1 1; 2 3 0 1; -1 1 2 2] (Observa  o: CONTEM
    AUTOVALORES IMAGINARIOS!!)
                                                (Autovalores: 4.56,
    1.89,   1  .23+0.833i,   1  .23-0.833i)
 A  =

   1.    0.    2.    0.
   2.    1.    1.    1.
   2.    3.    0.    1.
  -1.    1.    2.    2.

--> x0=[1;2;3;4]
 x0  =

   1.
   2.
   3.
   4.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   4.5594454
 x1  =

   0.5618853
   0.8475481
   1.
   0.8930305
 k  =

   29.
 n_erro  =

   8.890D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

  "O metodo convergiu com a precisao passada"
 lambda  =
```

```
   4.5594454
 x1  =

   0.3339143
   0.5036766
   0.5942749
   0.5307056
 k  =

   26.
 n_erro  =

   4.985D-11

-------------------------------------------------------------------

--> A=[0 1; 0 0] (Autovalor: 0) (Nao diagonalizavel)
 A  =

   0.   1.
   0.   0.

--> x0=[3;4]
 x0  =

   3.
   4.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =
```

```
   101.
 n_erro  =

   Nan


-----------------------------------------------------------------

--> A=[1 1; -1 3] (Autovalor: 2)
 A  =

   1.   1.
  -1.   3.

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

   1.
   2.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

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

   2.0196078
 x1  =

   1.9805825
   2.0194175
 k  =

   101.
 n_erro  =

   0.0001904

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

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

   2.0196060
 x1  =

   1.4141469
   1.4416061
 k  =

   101.
 n_erro  =

   0.0000952

-----------------------------------------------------------------

--> A=[2 1; 1 2] (Autovalores: 3, 1)
```

```
 A  =

   2.   1.
   1.   2.

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

   1.
   3.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

   1.
   1.
 k  =

   23.
 n_erro  =

   6.373D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.
 x1  =

   0.7071068
   0.7071068
 k  =

   21.
 n_erro  =

   9.560D-11
```

Através de alguns testes, foi possível notar que, como esperado, sendo A diagonalizável, o método irá convergir, caso contrário, é possível que o número de iterações passe do permitido e o lambda correto não seja encontrado. É possível notar também que os métodos podem não achar os autovalores corretos para matrizes com menos autovalores que sua ordem (por exemplo, matrizes 2x2 com um autovalor, 3x3 com dois autovalores, etc).

Testando os tempos para diferentes números de iterações com variáveis de entrada constantes:

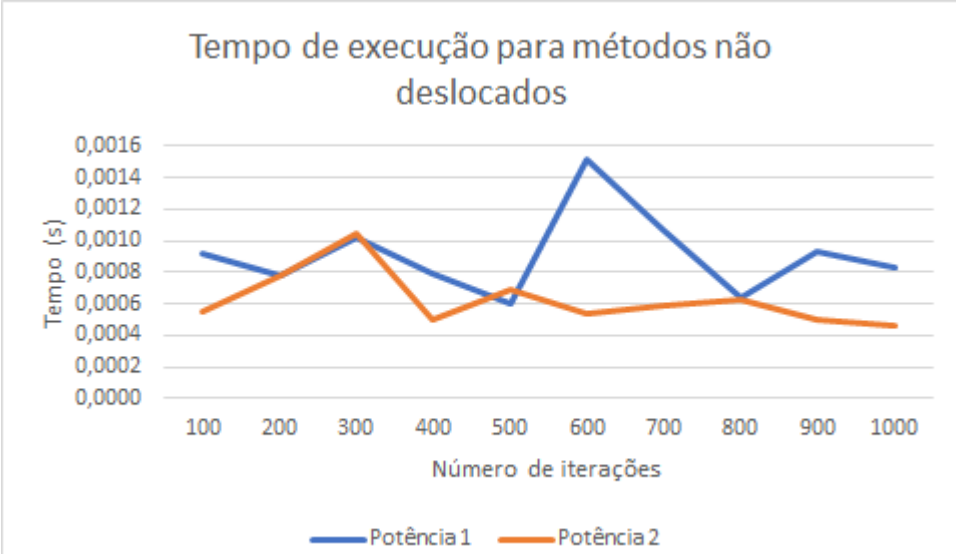
```
--> A=[-1 3 -1; -3 5 -1; -3 3 1]
```

```
 A  =

  -1.   3.  -1.
  -3.   5.  -1.
  -3.   3.   1.

--> x0=[1;3;5]
 x0  =

   1.
   3.
   5.

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)
    ;t1=toc()

  "O metodo convergiu com a precisao passada"
 t1  =

   0.0009175

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),200)
    ;t2=toc()

  "O metodo convergiu com a precisao passada"
 t2  =

   0.0007797

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),300)
    ;t3=toc()

  "O metodo convergiu com a precisao passada"
 t3  =

   0.0010204

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),400)
    ;t4=toc()

  "O metodo convergiu com a precisao passada"
 t4  =

   0.0007966

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),500)
    ;t5=toc()

  "O metodo convergiu com a precisao passada"
 t5  =

   0.0006009

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),600)
    ;t6=toc()

  "O metodo convergiu com a precisao passada"
```

```
 t6  =

   0.0015199

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),700)
    ;t7=toc()

  "O metodo convergiu com a precisao passada"
 t7  =

   0.0010745

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),800)
    ;t8=toc()

  "O metodo convergiu com a precisao passada"
 t8  =

   0.0006365

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),900)
    ;t9=toc()

  "O metodo convergiu com a precisao passada"
 t9  =

   0.0009277

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10)
    ,1000);t10=toc()

  "O metodo convergiu com a precisao passada"
 t10  =

   0.0008272

-------------------------------------------------------------

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)
    ;t11=toc()

  "O metodo convergiu com a precisao passada"
 t11  =

   0.0005534

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),200)
    ;t12=toc()

  "O metodo convergiu com a precisao passada"
 t12  =

   0.0007731

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),300)
    ;t13=toc()

```

```
 "O metodo convergiu com a precisao passada"
 t13  =

   0.0010449

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),400)
    ;t14=toc()

 "O metodo convergiu com a precisao passada"
 t14  =

   0.0005043

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),500)
    ;t15=toc()

 "O metodo convergiu com a precisao passada"
 t15  =

   0.0006954

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),600)
    ;t16=toc()

 "O metodo convergiu com a precisao passada"
 t16  =

   0.0005419

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),700)
    ;t17=toc()

 "O metodo convergiu com a precisao passada"
 t17  =

   0.0005887

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),800)
    ;t18=toc()

 "O metodo convergiu com a precisao passada"
 t18  =

   0.0006297

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),900)
    ;t19=toc()

 "O metodo convergiu com a precisao passada"
 t19  =

   0.0005034

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10)
    ,1000);t20=toc()

 "O metodo convergiu com a precisao passada"
```

```
 t20  =

   0.0004615
```



Tempo de execução dos métodos não deslocados para as entradas passadas, alterando M

Pela imagem, é possível ver que, mesmo que próximos, o Método da Potência 2 costuma ter tempos mais rápidos que o Método da Potência 1, já que calcular o quociente de Rayleigh é mais rápido que encontrar a coordenada de maior módulo no vetor.

3 Questão 4

Vamos usar novamente a matriz diagonalizável $A = [-1, 3, -1; -3, 5, -1; -3, 3, 1]$, cujos autovalores são $\lambda = 1, 2, 2$, para comparar o número de iterações necessárias e o tempo de execução. Para fazer da mesma maneira que foi feita anteriormente, usaremos $x0 = [1; 3; 5]$.

```
--> A=[-1 3 -1; -3 5 -1; -3 3 1] (Autovalores: 2, 1)
 A  =

  -1.    3.   -1.
  -3.    5.   -1.
  -3.    3.    1.

--> x0=[1; 3; 5]
 x0  =

```

```
   1.
   3.
   5.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   2.0000000
 x1  =

   0.3333333
   0.6666667
   1.
 k  =

   33.
 n_erro  =

   5.174D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   2.0000000
 x1  =

   0.2672612
   0.5345225
   0.8017837
 k  =

   30.
 n_erro  =

   8.147D-11

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)
    ; t1=toc()

 "O metodo convergiu com a precisao passada"
 t1  =

   0.0008487

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)
    ; t2=toc()

 "O metodo convergiu com a precisao passada"
 t2  =

   0.0004305
```

Usando essa matriz e autovetor, é possível ver que o algoritmo 1, que usa a

coordenada de maior módulo de $x1$ como $lambda$, foi feito com mais iterações e levou mais tempo, aproximadamente o dobro, que o algoritmo 2, que usa $lambda$ como o quociente de Rayleigh: $lambda = x0^T * x1$. Com isso, fica claro que usar o quociente de Rayleigh torna o processo mais simples e consequentemente rápido, já que o algoritmo não precisa encontrar a coordenada de maior módulo do autovetor a cada iteração.

4 Testando os métodos da Potência Deslocada (Questões 2 e 3)

Ambas as funções criadas para determinar o autovalor dominante da matriz A mais próximo do alfa passado através do Método da Potência Deslocada (Iteração Inversa e Método de Rayleigh) estão no arquivo "Metodo_da_Potencia_Deslocada.sci".

Testando com diferentes matrizes e vetores iniciais teremos:

```
--> A=[3 1; 1 3] (Autovalores: 4, 2)
 A  =

   3.   1.
   1.   3.

--> x0=[14;30]
 x0  =

   14.
   30.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,5,100)

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

   4.
 x1  =

   0.7071068
   0.7071068
 k  =

   101.
 n_erro  =

   2.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,4,100)

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

   Nan
```

```
 x1  =

   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,3,100)

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

   3.7664234
 x1  =

   0.4228855
   0.9061831
 k  =

   101.
 n_erro  =

   0.6834861

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,2,100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,1,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   2.
 x1  =

  -0.7071068
```

```
   0.7071068
 k  =

   23.
 n_erro  =

   5.842D-11

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,5,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   4.0000000
 x1  =

  -0.7071068
  -0.7071068
 k  =

   5.
 n_erro  =

   0.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,4,100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,3,100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =
```

```
   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,2,100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,1,100)

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

   Nan
 x1  =

   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

-------------------------------------------------------------

--> A=[1 2 0; 0 3 0; 2 -4 2] (Autovalores: 3, 2, 1)
 A  =

   1.   2.   0.
   0.   3.   0.
   2.  -4.   2.

--> x0=[3;4;5]
 x0  =

   3.
   4.
   5.
```

```
--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,5,100)

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

   3.
 x1  =

   0.4082483
   0.4082483
  -0.8164966
 k  =

   101.
 n_erro  =

   2.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,4,100)

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

   3.
 x1  =

   0.4082483
   0.4082483
  -0.8164966
 k  =

   101.
 n_erro  =

   2.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,3,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

```

```
--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,2,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,1,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,0,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   1.0000000
 x1  =

  -0.4472136
   2.776D-17
   0.8944272
 k  =

   35.
 n_erro  =

   6.403D-11

```

```
--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,5,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,4,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

  -0.4082483
  -0.4082483
   0.8164966
 k  =

   7.
 n_erro  =

   6.301D-14

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,3,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan
```

```
--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,2,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,1,100)

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

   Nan
 x1  =

   Nan
   Nan
   Nan
 k  =

   101.
 n_erro  =

   Nan

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,0,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   1.0000000
 x1  =

   0.4472136
   8.225D-23
  -0.8944272
 k  =

   7.
 n_erro  =

   7.206D-11

```

```
-------------------------------------------------------------

--> A=[0 2 -3 5; 2 4 1 3; -3 1 1 4; 5 3 4 2] (Autovalores: ~9.205,
    ~3.563, ~1.295, ~-7.063)
 A  =

   0.    2.   -3.    5.
   2.    4.    1.    3.
  -3.    1.    1.    4.
   5.    3.    4.    2.

--> x0=[1;2;3;4]
 x0  =

   1.
   2.
   3.
   4.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,10,100)

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

   9.2050871
 x1  =

   0.405511
   0.5832051
   0.2445709
   0.6600134
 k  =

   101.
 n_erro  =

   2.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,9,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   9.2050871
 x1  =

   0.405511
   0.5832051
   0.2445709
   0.6600134
 k  =

   8.
 n_erro  =
```

```
   4.979D-11

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,4,100)

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

   3.5634895
 x1  =

  -0.5648791
  -0.0809848
   0.812747
   0.117454
 k  =

   101.
 n_erro  =

   2.0000000

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,3,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   3.5634895
 x1  =

  -0.5648791
  -0.0809848
   0.812747
   0.117454
 k  =

   22.
 n_erro  =

   4.259D-11

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,2,100)

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

   1.2948071
 x1  =

   0.345427
  -0.8082635
   0.0919221
   0.4679109
 k  =

```

```
   101.
 n_erro  =

   2.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,1,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   1.2948071
 x1  =

   0.345427
  -0.8082635
   0.0919221
   0.4679109
 k  =

   13.
 n_erro  =

   1.179D-11

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,-7,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

  -7.0633837
 x1  =

  -0.6302005
   0.0048345
  -0.5207473
   0.5758873
 k  =

   7.
 n_erro  =

   1.691D-12

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,10,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   9.2050871
 x1  =

   0.405511
   0.5832051
   0.2445709
```

```
   0.6600134
 k  =

   4.
 n_erro  =

   6.460D-12

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,9,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   9.2050871
 x1  =

   0.405511
   0.5832051
   0.2445709
   0.6600134
 k  =

   4.
 n_erro  =

   1.665D-16

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,4,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.5634895
 x1  =

   0.5648791
   0.0809848
  -0.812747
  -0.117454
 k  =

   5.
 n_erro  =

   1.119D-16

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,3,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.5634895
 x1  =
```

```
   0.5648791
   0.0809848
  -0.812747
  -0.117454
 k  =

   5.
 n_erro  =

   1.127D-16

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,2,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.5634895
 x1  =

  -0.5648791
  -0.0809848
   0.812747
   0.117454
 k  =

   6.
 n_erro  =

   0.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,1,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   1.2948071
 x1  =

  -0.345427
   0.8082635
  -0.0919221
  -0.4679109
 k  =

   5.
 n_erro  =

   6.883D-11

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,-7,100)

  "O metodo convergiu com a precisao passada"
 lambda  =
```

```
  -7.0633837
 x1  =

   0.6302005
  -0.0048345
   0.5207473
  -0.5758873
 k  =

   5.
 n_erro  =

   1.118D-16
```

Usando métodos deslocados, é possível notar que não basta mais que a matriz seja diagonalizável para que o método convirja. No caso, é preciso que o alfa passado seja menor ou igual que os autovalores da matriz.

Observando os testes passados, é notável que se o alfa passado for igual ao autovalor da matriz, os métodos irão falhar. Contudo, é perceptível que o método de Rayleigh não funciona corretamente sempre, como é possível ver no último exemplo mostrado, em que o lambda esperado era -7.0633837 e o método devolveu 3.5634895.

Testando os tempos para diferentes $alfas$ com variáveis de entrada constantes:

```
--> A=[0 2 -3 5; 2 4 1 3; -3 1 1 4; 5 3 4 2] (Autovalores: ~9.205,
    ~3.563, ~1.295, ~-7.063)
 A  =

   0.   2.  -3.   5.
   2.   4.   1.   3.
  -3.   1.   1.   4.
   5.   3.   4.   2.

--> x0=[1;2;3;4]
 x0  =

   1.
   2.
   3.
   4.

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),10,100);t1=toc()

 "O numero de iteracoes passou do maximo de iteracoes permitido"
 t1  =

   0.0264886

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),9,100);t2=toc()

 "O metodo convergiu com a precisao passada"
 t2  =
```

```
28
29    0.0033274
30
31 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
      ,10^(-10),4,100);t3=toc()
32
33  "O numero de iteracoes passou do maximo de iteracoes permitido"
34  t3  =
35
36    0.0322190
37
38 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
      ,10^(-10),3,100);t4=toc()
39
40  "O metodo convergiu com a precisao passada"
41  t4  =
42
43    0.0105523
44
45 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
      ,10^(-10),2,100);t5=toc()
46
47  "O numero de iteracoes passou do maximo de iteracoes permitido"
48  t5  =
49
50    0.0255967
51
52 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
      ,10^(-10),1,100);t6=toc()
53
54  "O metodo convergiu com a precisao passada"
55  t6  =
56
57    0.0035725
58
59 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
      ,10^(-10),-7,100);t7=toc()
60
61  "O metodo convergiu com a precisao passada"
62  t7  =
63
64    0.0026467
65
66 -------------------------------------------------------------------
67
68 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
      ,10^(-10),10,100);t8=toc()
69
70  "O metodo convergiu com a precisao passada"
71  t8  =
72
73    0.0014118
74
75 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
      ,10^(-10),9,100);t9=toc()
76
77  "O metodo convergiu com a precisao passada"
```

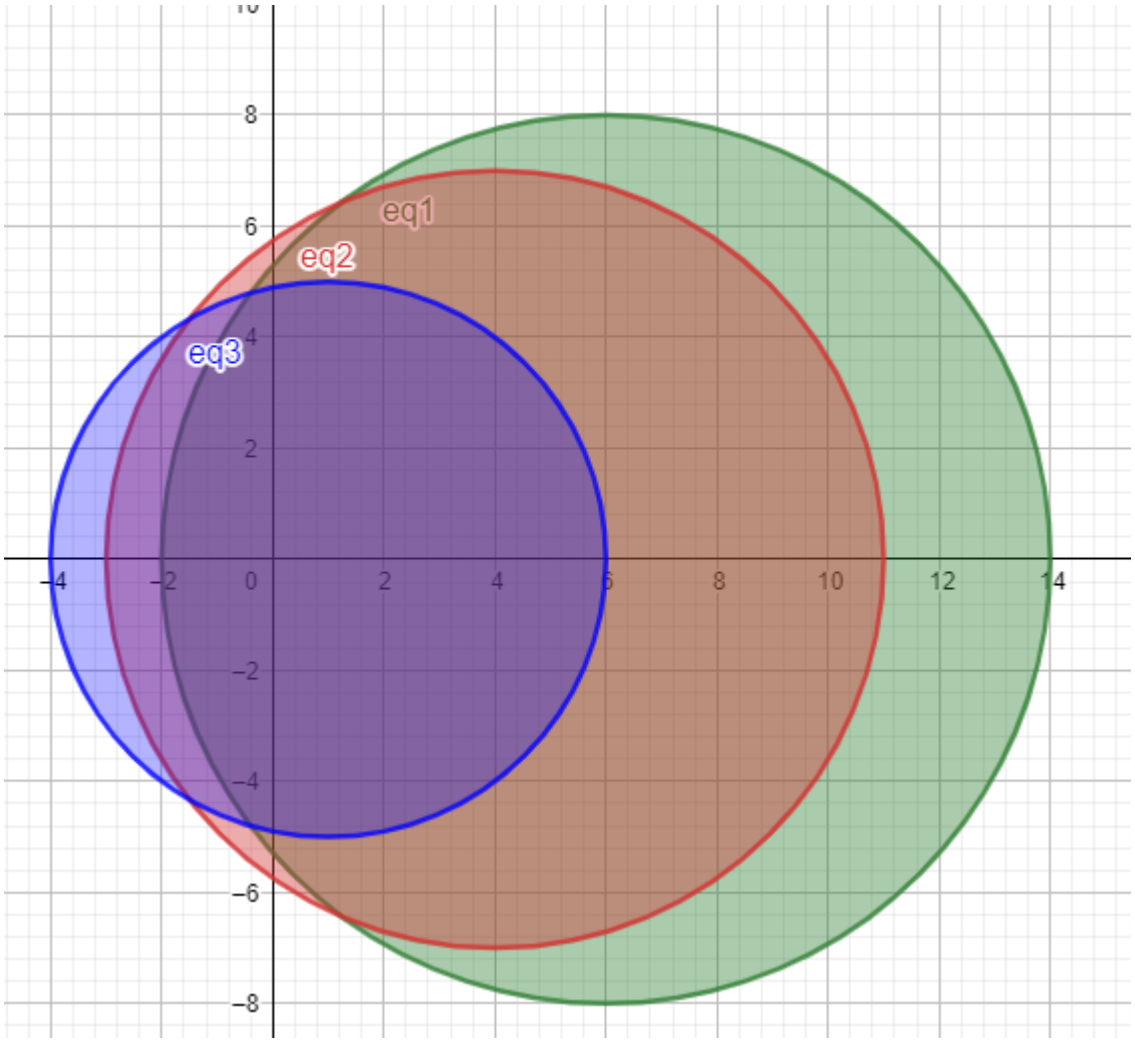
```
 t9  =

   0.0013488

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),4,100);t10=toc()

  "O metodo convergiu com a precisao passada"
 t10  =

   0.0021528

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),3,100);t11=toc()

  "O metodo convergiu com a precisao passada"
 t11  =

   0.0027536

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),2,100);t12=toc()

  "O metodo convergiu com a precisao passada"
 t12  =

   0.0018643

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),1,100);t13=toc()

  "O metodo convergiu com a precisao passada"
 t13  =

   0.0020644

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),-7,100);t14=toc()

  "O metodo convergiu com a precisao passada"
 t14  =

   0.0023858
```

Tempo de execução dos métodos deslocados para as entradas passadas, alterando $alfa$

Pelo gráfico, é possível ver que o algoritmo de Rayleigh foi bem mais rápido que o método deslocado inverso, embora o método inverso apresente tanto picos de 0,035 segundos quanto vales de aproximadamente 0,005, se aproximando do método de Rayleigh. Isso ocorre pois o algoritmo de Rayleigh atualiza o $alfa$ a cada iteração, e portanto consegue chegar mais perto do autovalor de forma mais rápida que o método deslocado inverso.

5 Questão 5

Para essa questão, vamos usar a matriz simétrica 3x3 $A = [1, 2, 3; 2, 4, 5; 3, 5, 6]$ e o autovetor $x0 = [1; 2; 3]$, para encontrar os discos de Gerschgorin dessa matriz, de forma a estimar a localização de seus autovalores. Pelo Teorema de Gerschgorin ($A = [a_{ij}]_{nxn} \Rightarrow r_i = \sum_{j \neq i} |a_{ij}|$), os autovalores dessa matriz estão localizados no eixo dos reais dentro de discos com centros em a_{ii}, ou seja, em 1, 4 e 6, e raios respectivamente iguais a 5, 7 e 8.



Discos de Gershgorin calculados

- Disco azul: $|z-6| \leq 8$
- Disco vermelho: $|z-4| \leq 7$
- Disco cinza: $|z-1| \leq 5$

```
--> 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,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,1,100)

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

   0.1709152
 x1  =

  -0.591009
   0.7369762
  -0.3279853
 k  =

   101.
 n_erro  =

   2.0000000

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),1,100); t1=toc()

  "O numero de iteracoes passou do maximo de iteracoes permitido"
 t1  =

   0.0219016

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,4,100)

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

   0.1709152
 x1  =

  -0.5910089
```

```
   0.7369763
  -0.3279854
 k  =

   101.
 n_erro  =

   2.0000000

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),4,100); t2=toc()

 "O numero de iteracoes passou do maximo de iteracoes permitido"
 t2  =

   0.0173268

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,6,100)

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

   11.344814
 x1  =

   0.327988
   0.5910056
   0.7369778
 k  =

   101.
 n_erro  =

   0.0000097

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),6,100); t3=toc()

 "O numero de iteracoes passou do maximo de iteracoes permitido"
 t3  =

   0.0178859

-------------------------------------------------------------------

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,1,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   11.344814
 x1  =

   0.3279853
   0.591009
```

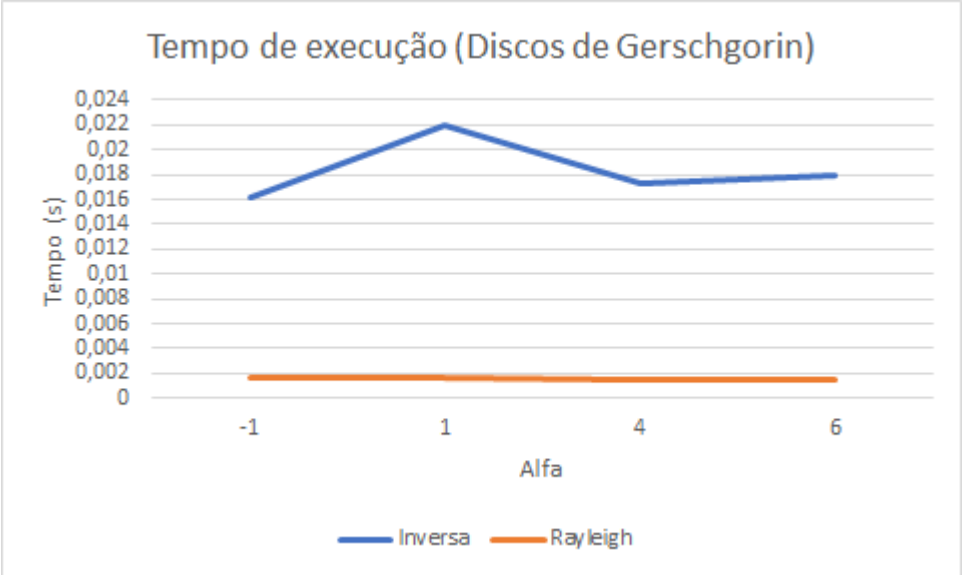
```
   0.7369762
 k  =

   7.
 n_erro  =

   0.

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),1,100); t4=toc()

  "O metodo convergiu com a precisao passada"
 t4  =

   0.0016335

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,4,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   11.344814
 x1  =

  -0.3279853
  -0.591009
  -0.7369762
 k  =

   5.
 n_erro  =

   0.

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),4,100); t5=toc()

  "O metodo convergiu com a precisao passada"
 t5  =

   0.0014949

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,6,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   11.344814
 x1  =

   0.3279853
   0.591009
   0.7369762
 k  =
```

```
   5.
 n_erro  =

   0.

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),6,100); t6=toc()

  "O metodo convergiu com a precisao passada"
 t6  =

   0.0015383
```

Calculando o polinômio característico dessa matriz, chegamos a $-\lambda^3 + 11\lambda^2 + 4\lambda - 1 = 0$, e fazendo essa conta, achamos $\lambda_1 = 11.344814282762078, \lambda_2 = 0.17091518882718093$ e $\lambda_3 = -0.5157294715892573$. Sendo assim, é possível ver que os métodos deslocados foram boas aproximações para achar os autovalores da matriz, com exceção do autovalor negativo, que não foi encontrado. Assim, usaremos um $alfa$ diferente de a_{ii} para tentar encontrar o autovalor restante:

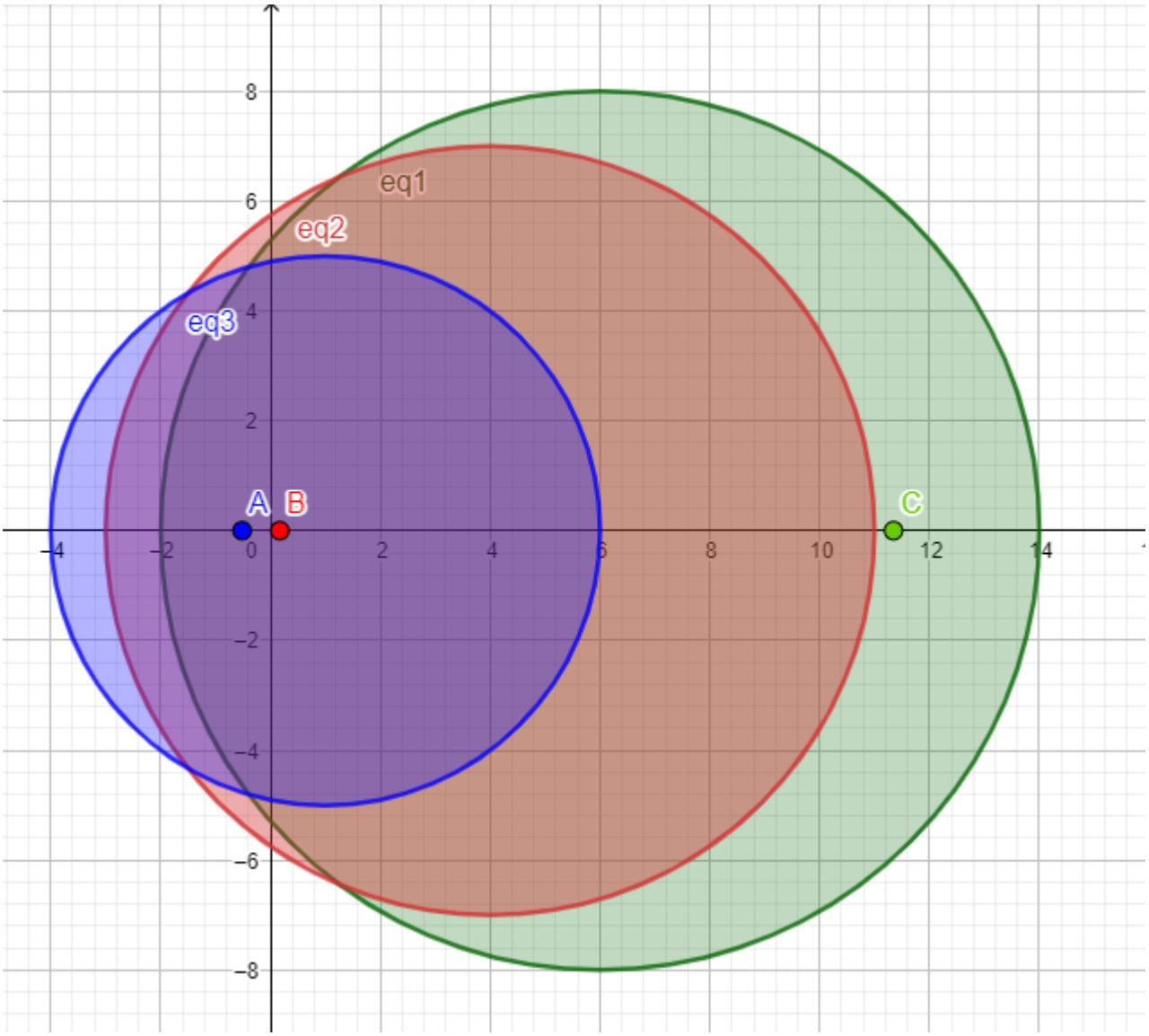
```

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,-1,100)

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

  -0.5157295
 x1  =

   0.7369762
   0.3279853
  -0.591009
 k  =

   101.
 n_erro  =

   2.

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),-1,100); t7=toc()

  "O numero de iteracoes passou do maximo de iteracoes permitido"
 t7  =

   0.0162406

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,-1,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

  -0.5157295
 x1  =

```

```
   0.7369762
   0.3279853
  -0.591009
 k   =

   6.
 n_erro   =

   4.996D-16

--> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),-1,100); t8=toc()

  "O metodo convergiu com a precisao passada"
 t8   =

   0.0016906
```



Tempos de execução nos Discos de Gershgorin

Mesmo que o método de Rayleigh tenha devolvido o mesmo *lambda* para diferentes *alphas*, o algoritmo de Rayleigh foi bem mais rápido que o método deslocado inverso, aproximadamente 10 vezes mais rápido. Isso ocorre pois o algoritmo de Rayleigh atualiza o $alfa$ a cada iteração, e portanto consegue chegar mais perto do autovalor de forma mais rápida que o método deslocado inverso.



Discos de Gershgorin calculados com os autovalores indicados

6 Testes extras

Embora sabendo que sendo diagonalizável a matriz irá convergir, o oposto não necessariamente é verdade. Por exemplo, buscando matrizes para testar, encontrei uma matriz não diagonalizável que, contrariando a minha expectativa, passou nos testes de convergência do método:

```
--> A=[2 1 1; 0 2 2; 0 0 3]
 A  =

   2.    1.    1.
   0.    2.    2.
   0.    0.    3.

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

   1.
   2.
   3.
```

$Det(A - \lambda * I) = (2 - \lambda)^2 * (3 - \lambda) = 0 \Rightarrow \lambda_1 = 3, \lambda_2 = 2, \lambda_3 = 2$

```
--> D=[2 0 0; 0 2 0; 0 0 3]
 D  =

   2.    0.    0.
   0.    2.    0.
   0.    0.    3.
```

Para $\lambda = 2$:

```
--> v=[a;b;c]

v  =

   a
   b
   c

--> A_linha=[0 1 1; 0 0 2; 0 0 1]
 A_linha  =

   0.    1.    1.
   0.    0.    2.
   0.    0.    1.

A_linha*v=[0;0;0]

a=a
b=0
c=0
```

Com isso, temos que não existem v_1, v_2 LI tal que $Av_1 = 2v_1$ e $Av_2 = 2v_2$.

Tomando $AP = P * [2, 0, 0; 0, 2, 0; 0, 0, 3]$, podemos fazer $v_1 = P.e_1 \Rightarrow Av_1 = AP.e_1 = P * [2, 0, 0; 0, 2, 0; 0, 0, 3] * [1; 0; 0] = P.2e_1 = 2v_1$ e $v_2 = P.e_2 \Rightarrow Av_2 = AP.e_2 = P * [2, 0, 0; 0, 2, 0; 0, 0, 3] * [0; 1; 0] = P.2e_2 = 2v_2$. Com isso, teríamos que ter v_1 e v_2 LI já que P é inversível, porém já havíamos provado anteriormente que não existem v_1 e v_2 LI, portanto A não pode ser diagonalizável. Ainda assim, é possível que A seja usado no método:

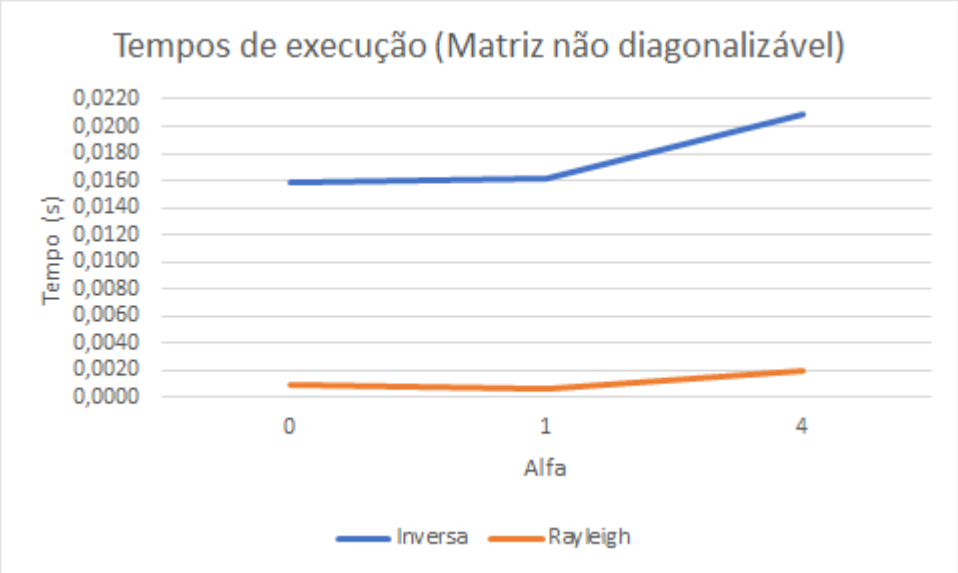
```
--> A=[2 1 1; 0 2 2; 0 0 3] (Autovalores: 3, 2, 2)
 A  =
```

```
 
   2.    1.    1.
   0.    2.    2.
   0.    0.    3.

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

   1.
   2.
   3.

--> [lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

   1.
   0.6666667
   0.3333333
 k  =

   63.
 n_erro  =

   7.924D-11

--> [lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

   0.8017837
   0.5345225
   0.2672612
 k  =

   60.
 n_erro  =

   8.488D-11

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,0,100)

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

   1.9791757
 x1  =

```

```
   0.9997831
  -0.0208288
   3.842D-20
 k  =

   101.
 n_erro  =

   0.0002192

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,0,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

   0.8017837
   0.5345225
   0.2672612
 k  =

   3.
 n_erro  =

   2.483D-16

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,1,100)

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

   1.9897970
 x1  =

   0.9999479
  -0.0102036
   6.037D-33
 k  =

   101.
 n_erro  =

   0.0001052

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,1,100)

 "O metodo convergiu com a precisao passada"
 lambda  =

   1.8571429
 x1  =

```

```
   0.2672612
  -0.5345225
   0.8017837
 k  =

   1.
 n_erro  =

   2.544D-16

--> [lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0,10^(-10)
    ,4,100)

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

   3.
 x1  =

   0.8017837
   0.5345225
   0.2672612
 k  =

   101.
 n_erro  =

   2.

--> [lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0,10^(-10)
    ,4,100)

  "O metodo convergiu com a precisao passada"
 lambda  =

   3.0000000
 x1  =

  -0.8017837
  -0.5345225
  -0.2672612
 k  =

   7.
 n_erro  =

   2.654D-13

-------------------------------------------------------------------

--> tic();[lambda,x1,k,n_erro]=Metodo_potencia_1(A,x0,10^(-10),100)
    ; t1=toc()

  "O metodo convergiu com a precisao passada"
 t1  =

   0.4845251
```

```
167
168 --> tic();[lambda,x1,k,n_erro]=Metodo_potencia_2(A,x0,10^(-10),100)
    ; t2=toc()
169
170  "O metodo convergiu com a precisao passada"
171 t2  =
172
173    0.0013723
174
175 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),4,100); t3=toc()
176
177  "O numero de iteracoes passou do maximo de iteracoes permitido"
178 t3  =
179
180    0.0208992
181
182 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),4,100); t4=toc()
183
184  "O metodo convergiu com a precisao passada"
185 t4  =
186
187    0.0019363
188
189 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),1,100); t5=toc()
190
191  "O numero de iteracoes passou do maximo de iteracoes permitido"
192 t5  =
193
194    0.0162066
195
196 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),1,100); t6=toc()
197
198  "O metodo convergiu com a precisao passada"
199 t6  =
200
201    0.0006033
202
203 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_inversa(A,x0
    ,10^(-10),0,100); t7=toc()
204
205  "O numero de iteracoes passou do maximo de iteracoes permitido"
206 t7  =
207
208    0.0159103
209
210 --> tic();[lambda,x1,k,n_erro]=Potencia_deslocada_Rayleigh(A,x0
    ,10^(-10),0,100); t8=toc()
211
212  "O metodo convergiu com a precisao passada"
213 t8  =
214
215    0.0009824
```



Tempos de execução de métodos deslocados em matriz não diagonalizável

Embora a matriz tenha passado no teste, é notável que os métodos deslocados não devolveram sempre o *lambda* esperado, especialmente para $\lambda = 2$. De qualquer forma, é visível que mesmo para uma matriz não diagonalizável, o Método da Potência 2 foi bem mais rápido que o 1, assim como o de Rayleigh foi bem mais rápido como o método inverso, como pode ser visto no gráfico.

7 Fontes de consulta

- Aulas Gravadas
- Álgebra Linear, David Poole (2^a edição)
- Álgebra Linear I - Aula 19, PUC RIO (http://www.mat.puc-rio.br/cursos/MAT1200/roteiros/rotaula19.pdf)
- Geogebra (para construção dos Discos de Gerschgorin)
- Exemplos de Matrizes 3x3 não diagonalizáveis, Matemática Universitária, (https://www.youtube.com/watch?v=ae1MZN-8SIY)
- Análise Numérica para Busca de Autovalores, Hortênsia Virginia Américo (https://repositorio.ufmg.br/bitstream/1843/EABA-978HQW/1/monografia_hortensia_americo.pdf)