Scientific Computing Assignment 2

Odysseas Lazaridis

September 2022

1 a(1)

Gershgorin's theorem indicates that the eigenvalues exist in an conjunction of disks who's centers are the diagonal elements of the matrix and the radii arise form the sum of the elements of each row minus the diagonal element. The code is shown below

```
def gershgorin(A):
    m, n = np.shape(A)
    centers = np.diagonal(A)
    radii = np.zeros(n)
    for i in range(n):
        radii[i] = np.abs(np.sum(A[i,:]) - centers[i])
    return centers, radii
```

2 a(2)

For the K matrix I found the following centers and corresponding their radii

Center	129292	103041	64967	43612	36273	37990	24166	11651	13865	5600
Radius	47747	22332	7943	6365	36273	2685	6552	609	227	4005
Center	1173	1760	288.42	86.897	13.893					
radius	22.719	150.93	68.784	1.4641	0.2056					

The combined area of the disks is depicted on figure 1

3 b(1)

The function lambda = rayleigh qt(A,x) just solves the system for λ assuming that A has a tranapose.

```
def rayleigh_qt(A,x):
lamda = (x.T@A@x)/(x.T@x)
return lamda
```

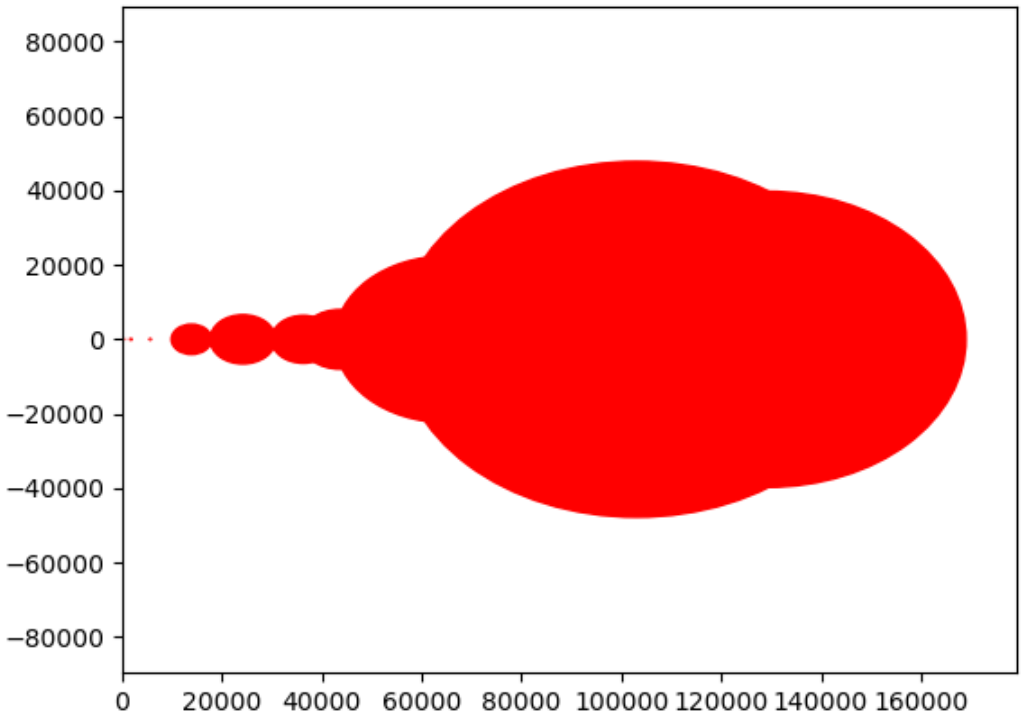

Figure 1: Combined area of the disks

4 b(2)

This function applies the matrix on an initial vector and after applying it enough times we can get as close as we want to the eigenvector.

```
def power_iterate(A,x0):
k=0
temp=10
while np.linalg.norm(temp-x0)>0.001:
    temp = x0
    x0 = A@x0
    x0 = x0/np.linalg.norm(x0)
    k+=1

return x0, k
```

5 b(3)

Applying the functions mentioned above on the Test Matrices we get the following we get the following eigenvectors (and the iteration that took in order to reach 0.001 to the eigenvectors).

Matrix	Eigenvector	iterations
A1	(0.707, 0.707)	2
A2	(0.707, 0.707)	2
A3	(0.302, 0.611,0.731)	7
A4	(0.234, 0.525, 0.819)	4
A5	(0.108, 0.253, 0.398, 0.542, 0.687)	4
A6	(0.455, 0.262, 0.623, 0.447, 0.368)	7

6 b(4)

Again, applying the same method to K matrix we get the following eigenvector

$$y = \begin{bmatrix} 8.19008 \times 10^{-1} \\ -5.6890 \times 10^{-1} \\ 7.1647 \times 10^{-2} \\ -1.2798 \times 10^{-2} \\ -4.5634 \times 10^{-03} \\ -1.6124 \times 10^{-2} \\ 1.3400 \times 10^{-4} \\ 4.1185 \times 10^{-4} \\ 2.8968 \times 10^{-4} \\ -4.9258 \times 10^{-5} \\ 3.6929 \times 10^{-6} \\ -8.9443 \times 10^{-8} \\ -1.5876 \times 10^{-10} \\ -9.3483 \times 10^{-13} \end{bmatrix} \tag{1}$$

Which corresponds to the eigenvalue 151361 as we can calculate using the Raylight quotient. This eigenvalue is the largest one because by applying the Matrix a lot of times, the only eigenvalue that survives is the largest one as we can see from below. The initial vector that we use x_0 can be written as

$$x_0 = \sum_{j=1}^{m} \alpha_j \upsilon_j \tag{2}$$

Our final approximated eigenvector will be expressed:

$$x_\kappa = Ax_{k-1} = A^2 x_{k-2} = ... = A^k x_0 \tag{3}$$

But x_0 is already known from eq. 2 so

$$x_k = A^k \sum_{j=1}^{n} (\alpha_j \upsilon_j = \sum_{j=1}^{n} \alpha_j A^k \upsilon_j = \sum_{j=1}^{n} \lambda_j^k \alpha_j \upsilon_j = l_1^k (\alpha_1 \upsilon_1 + \sum_{j=2}^{n} (\frac{\lambda_j}{\lambda_1})^k \alpha_j \upsilon_j) \tag{4}$$

Where for big values of k λ_1 is the only value that stays alive.

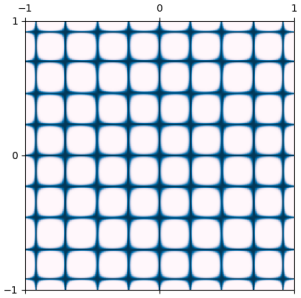

Figure 2: Nodes

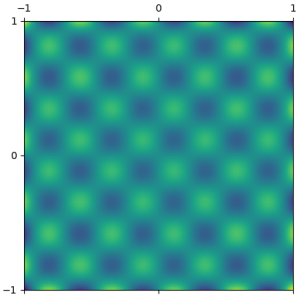

Figure 3: Waves

7 c(2)

Inverse power iteration is able to find the eigenvector which is closest to the approximation and not just the largest one. That's why here in order to find more eigenvalues I used different points on the area of the disk in order to maximize the possibility to find different eigenvalues. For every center I added and subtracted a radius and used all of these values as approximations. I did not optimize the code in this part because of lack of time. Some of the eigenvalues that I found have 0 sig figs on the residuals so I don't count them as found eigenvalues.

A1		
Theoretical	4	-2
Found	4	-2

A2		
Theoretical	4	22
Found	4	2

A3			
Theoretical	12.2990	-4.4805	-0.9585
Found	12.299	-4.4806	-0.9956

A4			
Theoretical	16.117	-1.117	0
Found	16.1168	-1.1168	0

A5					
Theoretical	68.6421	-3.6421	0	0	0
Found	68.6421	-	0		

I was able to find most of the eigenvalues where the significant digits of the residuals can be seen in the output of the code

A6					
Theoretical	2	2	2	1	1
Found	2	-	-	1.05	1.08

8 d(1)

The way we calculated the eigenvalues up until now is we started from a 'random' point and we hoped that it is close to the eigenvector we haven't calculated. With this method when we have very large matrices it becomes impossible to calculate in a finite amount of time.

9 d(2)

By using the same method that I used in c(2)

Eigenvalues found
129292
92848
52107
35893
35906
35897
21963
11632
13332
5557
1168
1798
286
86.9
13.9
13.9
113476

It seems that with this way there has been found 14/15 of the eigenvalues. I say 15 because this is the dimension of my matrix, so the max number of eigenvalues is 15. The eigenvectors of those can be seen on the output of the code.

We can see that there are a lot of multiples in the results and it is totally expected because as I mentioned before, as the matrix gets bigger it is harder to find for every real eigenvector an approximation which is closest to it than any other eigenvector