Tarea 10 Métodos Numéricos

Carlos Giovanny Encinia González

10 de octubre 2021

Desarrollo

M eigenvalores menores con el método de la inversa

Para poder aplicar este método primero debemos asegurarnos que la matriz con la que vamos a trabajar debe ser simétrica.

Ahora partimos del supuesto de que conocemos el eigenvalor menor λ_1 y entonces podemos escribir que un vector cualquiera puede ser escrito como una combinación de los eigenvectores:

$$v = \sum_{i=2}^{n} a_i x_i + a_1 x_1 \tag{1}$$

Multiplicando por l traspuesta del eigenvector dominante tenemos que:

$$x_1^T v = x_1^T \sum_{i=2}^{n} a_i x_i + a_1 x_1^T x_1 \tag{2}$$

Como la matriz es simétrica los eigenvectores son ortogonales y la sumatoria al tener elementos distintos a x_1 estos se vuelven 0 y nos queda que:

$$x_1^T v = a_1 \tag{3}$$

Para calcular el eigenvalor 2, simplemente debemos de quitar el componente asociado al eigenvalor dominante y tenemos que

$$\hat{v} = v - a_1 x_1 \tag{4}$$

Y sustituyendo el valor encontrado de a_1 tenemos que:

$$\hat{v_0} = v - x_1^T v x_1 \tag{5}$$

Donde $\hat{v}$ es el vector el cual utilizaremos en el algoritmo de la potencia para calcular el i eigenvector dominante.

Siguiendo esta lógica podemos encontrar los n valores dominantes, siempre restando el eigenvalor dominante anterior.

en general tendríamos que:

$$\hat{v_0} = v_0 - \sum_{i=1}^{n} x_i^T v_0 x_i \tag{6}$$

Con i=1 aumentando hasta hasta n
En seguida se muestra el pseudocódigo.

Algorithm 1: M eigenvalores menores

```
1  eigen_valor_vector_dominante (matrix, rows, x0);
   Input  : matrix: matriz que contiene coeficientes ecuación, x0: es el
            vector que se propone para solucionar, rows: numero de filas
   Output: x1, λ
2  i ← 1 for i:m do
3      ERROR = 1E^-9;
4      matrix ← readMatrix();
5      x0 ← creaVectorZeros(rows);
6      m ← rows;
7      x0_0 ← 1/sqrt(n);
8      iteration ← 0;
9      condition ← True λ_old ← 0;
10     while condition and iteration < LIMIT do
11         if i ≠ 0 then
12             for k ← 0, k < i, k++ do
13                 an = dot_vector(x0, xn_k, m);
14                 for j ← 0, j < m, j++ do
15                     x0_j ← x0_j - an * xn_{k,j};
16                 end
17             end
18         end
19         x1 = SolveLU(matrix, x0);
20         λ = productoPunto(x1^T, x0)/productoPunto(x0^T, x0);
21         if abs(λ_old − λ) < ERROR then
22             xn_i, vector_lambda_i ← x1, λ;
23             continue;
24         end
25         λ_old ← λ;
26         x0 ← x1;
27         x0 ← normalizar_vector(x0);
28         iteration ← iteration + 1;
29     end
30 end
```

Método de Jacobi

El método de Jacobi consiste en aplicar rotaciones estratégicas en ciertos puntos de la matriz cuadrada para diagonalizar la matriz, estos puntos se eligen tomando en cuenta el elemento que tiene el mayor valor absoluto fuera de la diagonal, es decir que se realiza un pivoteo.

En seguida se describen los detalles para llegar a su implementación algorítmica.

Primero para que el método converja necesitamos tener una matriz que sea simétrica y que los valores que la dimensión de la matriz esté alrededor de 1000.

Así que definimos:

$$A = A^T \tag{7}$$

Recordando la definición de valores propios tenemos que:

$$Ax = \lambda x \tag{8}$$

Si juntamos todos los eigenvalores en una matriz diagonal *Lambda* y todos los eigenvalores los reunimos como columnas en una matriz Φ

Entonces podemos expresar el problema de eigenvalores como

$$A\Phi = \Phi\Lambda \tag{9}$$

Teniendo en cuenta que los eigenvectores de una matriz simétrica son ortogonales podemos multiplicar por la traspuesta de Φ a la izquierda y nos queda que:

$$\Phi^T A\Phi = \Lambda \tag{10}$$

Esto no nos ayuda a encontrar los eigenvalores pero podemos proponer una serie de vectores que estarán rotando a la matriz A esto con el fin de poder diagonalizarla y de esta manera poder encontrar la respuesta a nuestro problema.

Entonces si hacemos:

$$T_m^T \cdots T_2^T \cdot T_1^T \cdot T_0^T \cdot A \cdot T_1 \cdot T_1 \cdot T_2 \cdots T_m = \Lambda \tag{11}$$

Ahora debido a que son rotaciones, proponemos nuestro cambio de variables como:

$$x = x'cos(\theta) - y'sen(\theta) \tag{12}$$

$$x = x'sen(\theta) + y'cos(\theta) \tag{13}$$

O podemos escribir que:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} cos(\theta) & -sen(\theta) \\ sen(\theta) & cos(\theta) \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \tag{14}$$

Esto aplicado a nuestra matriz quiere decir que primero debemos de encontrar el valor absoluto mas grande fuera de la diagonal y una vez encontrado debemos basarnos en sus índices, para poder crear una matriz de rotación. cuyos elementos serán como se muestra a continuación.

Sea T_0 la matriz de rotación.

$$T_{i,i} = cos(\theta) \tag{15}$$

$$T_{i,j} = -sen(\theta) \tag{16}$$

$$T_{j,i} = sen(\theta) \tag{17}$$

$$T_{j,j} = cos(\theta) \tag{18}$$

Los demás elementos de la diagonal serán 1, y cualquier otro elemento de la matriz es igual a 0.

Donde i, j son los índices correspondientes a el elemento encontrado en la matriz A.

Para una primera rotación tendremos:

$$T_0^T \cdot A \cdot T_0 \tag{19}$$

Primero proponemos una matriz B tal que:

$$B = A \cdot T_0 \tag{20}$$

Y

$$C = T_0^T \cdot B \tag{21}$$

Ahora para visualizar el resultado de B podemos escribir:

$$B = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & a_{3,m} \\ a_{m,1} & a_{m,2} & \cdots & a_{m,m} \end{pmatrix} \begin{pmatrix} cos(\theta) & 0 & \cdots & -sen(\theta) \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ sen(\theta) & 0 & \cdots & cos(\theta) \end{pmatrix} \tag{22}$$

Realizando las multiplicaciones obtenemos que:

$$B = \begin{pmatrix} a_{1,1}cos(\theta) + a_{1,m}sen(\theta) & a_{1,2} & \cdots & a_{1,m}cos(\theta) - a_{1,1}sen(\theta) \\ a_{2,1}cos(\theta) + a_{2,m}sen(\theta) & a_{2,2} & \cdots & a_{2,m}cos(\theta) - a_{2,1}sen(\theta) \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}cos(\theta) + a_{m,m}sen(\theta) & a_{m,2} & \cdots & a_{m,m}cos(\theta) - a_{m,1}sen(\theta) \end{pmatrix} \tag{23}$$

En resumen observamos que lo único que cambia son las filas que son mapeadas debido a los índices del elemento con valor absoluto menor fuera de la diagonal.

En general podemos escribir

$$b_{l,i} = a_{l,i}cos(\theta) + a_{l,j}sen(\theta) \tag{24}$$

$$b_{l,j} = -a_{l,i}sen(\theta) + a_{l,j}cos(\theta) \tag{25}$$

donde

$$1 \leq l \leq m \tag{26}$$

E i y j, son índices correspondientes al elemento con mayor valor absoluto fuera de la diagonal.

Ahora para visualizar el resultado de C tenemos que:

$$C = \begin{pmatrix} cos(\theta) & 0 & \cdots & sen(\theta) \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ -sen(\theta) & & \cdots & cos(\theta) \end{pmatrix} \begin{pmatrix} b_{1,i} & a_{1,2} & \cdots & b_{1,j} \\ b_{2,i} & a_{2,2} & \cdots & b_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m,i} & a_{m,2} & \cdots & b_{m,j} \end{pmatrix} \tag{27}$$

Multiplicando las matrices nos queda que

$$C = \begin{pmatrix} b_{1,i}cos(\theta) + b_{m,i}sen(\theta) & a_{1,2}cos(\theta) + a_{m,2}sen(\theta) & \cdots & b_{1,j}cos(\theta) + b_{m,j}sen(\theta) \\ b_{2,i} & a_{2,2} & \cdots & b_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ -b_{1,i}sen(\theta) + b_{m,i}cos(\theta) & -a_{1,2}sen(\theta) + a_{m,2}cos(\theta) & \cdots & -b_{1,j}sen(\theta) + b_{m,j}cos(\theta) \end{pmatrix} \tag{28}$$

Observando la matriz podemos reescribir los resultados en términos de los elementos de la matriz A tomando en cuenta lo siguiente

en nuestro caso sea $i = 1$ y $j = m$

Para $c_{i,i}$ tenemos que:

$$c_{i,i} = b_{i,i}cos(\theta) + b_{j,j}sen(\theta) \tag{29}$$

Sustituyendo lo obtenido en la ecuación (24) nos queda que

$$c_{i,i} = [a_{i,i}cos(\theta) + a_{i,j}sen(\theta)]cos(\theta) + [a_{j,i}cos(\theta) + a_{j,j}sen(\theta)]sen(\theta) \tag{30}$$

multiplicando y reagrupando nos queda que:

$$\boxed{c_{i,i} = a_{i,i}cos^2(\theta) + 2a_{i,j}sen(\theta)cos(\theta) + a_{j,j}sen^2(\theta)} \tag{31}$$

Ahora para calcular $c_{j,j}$ tenemos que

$$c_{j,j} = -b_{i,j}sen(\theta) + b_{j,j}cos(\theta) \tag{32}$$

sustituyendo los resultados de la ecuación (25) tenemos que:

$$c_{j,j} = -[-a_{i,i}sen(\theta) + a_{i,j}cos(\theta)]sen(\theta) + [-a_{j,i}sen(\theta) + a_{j,j}cos(\theta)]cos(\theta) \tag{33}$$

Multiplicando y agrupando nos queda que:

$$\boxed{c_{j,j} = a_{i,i}sen^2(\theta) - 2a_{i,j}sen(\theta)cos(\theta) + a_{j,j}cos^2(\theta)} \tag{34}$$

Ahora analizaremos los términos ubicados en i, j

$$c_{i,j} = -b_{i,i}sen(\theta) + b_{j,i}cos(\theta) \tag{35}$$

Utilizando la ecuación (24) tenemos que:

$$c_{i,j} = -[a_{i,i}cos(\theta) + a_{i,j}sen(\theta)]sen(\theta) + [a_{j,i}cos(\theta) + a_{j,j}sen(\theta)]cos(\theta) \tag{36}$$

Multiplicando y agrupando términos semejantes nos queda que:

$$\boxed{c_{i,j} = sen(\theta)cos(\theta)[a_{j,j} - a_{i,i}] + a_{i,j}[cos^2(\theta) - sen^2(\theta)]} \tag{37}$$

Ahora para calcular los términos en $c_{j,i}$ tenemos que:

$$c_{j,i} = b_{i,j}cos(\theta) + b_{j,j}sen(\theta) \tag{38}$$

Utilizando la ecuación (25) tenemos que:

$$c_{j,i} = [-a_{i,i}sen(\theta) + a_{i,j}cos(\theta)]cos(\theta) + [-a_{j,i}sen(\theta) + a_{j,j}cos(\theta)]sen(\theta) \quad (39)$$

Multiplicando y agrupando términos semejantes nos queda que:

$$\boxed{c_{i,j} = sen(\theta)cos(\theta)[a_{j,j} - a_{i,i}] + a_{i,j}[cos^2(\theta) - sen^2(\theta)]} \quad (40)$$

Y observamos que

$$\boxed{c_{i,j} = c_{j,i}} \quad (41)$$

Ahora para los términos $c_{i,k}$ tenemos que:

$$\boxed{c_{i,k} = a_{i,k}cos(\theta) + a_{j,k}sen(\theta)} \quad (42)$$

Y además podemos observar que

$$\boxed{c_{i,k} = c_{k,i}} \quad (43)$$

Y se cumple para $k \neq i, j$
Ahora para calcular el termino $c_{j,k}$

$$\boxed{c_{j,k} = -a_{i,k}sen(\theta) + a_{j,k}cos(\theta)} \quad (44)$$

Y además podemos observar que

$$\boxed{c_{j,k} = c_{k,j}} \quad (45)$$

Y se cumple para $k \neq i, j$
Por ultimo nos quedan los términos $c_{k,l}$

$$\boxed{c_{k,l} = a_{k,l} = a_{l,k}} \quad (46)$$

Para $1 \leq k, l \leq m$ y $k, l \neq i, j$

Los índices i y j, son índices correspondientes al elemento con mayor valor absoluto fuera de la diagonal.

Ahora queremos hacer el elemento $c_{i,j}$ cero, esto para diagonalizar la matriz A, entonces debemos buscar un θ que nos lleve a este resultado.

Entonces de la ecuación (40) podemos aplicar propiedades trigonométricas y obtener que:

$$c_{i,j} = \frac{sen(2\theta)}{2}[a_{j,j} - a_{i,i}] + a_{i,j}cos(2\theta) \quad (47)$$

Haciendo $c_{i,j} = 0$ nos queda que:

$$0 = \frac{sen(2\theta)}{2}[a_{j,j} - a_{i,i}] + a_{i,j}cos(2\theta) \quad (48)$$

Pasando el segundo termino a la izquierda de la igualdad

$$-a_{i,j}cos(2\theta) = \frac{sen(2\theta)}{2}[a_{j,j} - a_{i,i}] \quad (49)$$

Ahora despejando las funciones seinodales, nos queda que:

$$-\frac{2a_{i,j}}{a_{j,j}-a_{i,i}} = \frac{sen(2\theta)}{cos(2\theta)} \tag{50}$$

Usando la propiedad de la $tan(\omega)$ nos queda que

$$\boxed{tan(2\theta) = \frac{2a_{i,j}}{a_{i,i}-a_{j,j}}} \tag{51}$$

$$\boxed{\theta = \frac{1}{2} arctan\left(\frac{2a_{i,j}}{a_{i,i}-a_{j,j}}\right)} \tag{52}$$

Y si $a_{j,j} = a_{i,i}$ sabemos que

$$\boxed{\theta = \frac{\pi}{4}} \tag{53}$$

Para calcular la matriz con los eigenvectores lo que podemos hacer es simplemente proponer una aproximación a la matriz con eigenvectores como la primera matriz de rotación, y esta se actualiza multiplicando con las siguientes matrices de rotación, esto se puede simplificar si utilizamos la ecuación (24) y (25) para calcular los nuevos elementos de nuestra matriz.

De la ecuación (11) definimos

$$\Phi = T_0 \cdot T_1 \cdot T_2 \cdots T_m \tag{54}$$

En seguida se muestra el pseudocódigo.

Algorithm 2: Mayor absoluto

```
1  mayor_absoluto (matrix, m);
   Input  : matrix: matriz que contiene coeficientes ecuación, m: numero
            de filas
   Output: i, j
2  mayor = abs(matrix_{0,1});
3  for k ← 0 : m do
4      for l ← 0 : m do
5          if k ≠ l then
6              if abs(matrix_{k,l} ≥ mayor then
7                  i ← l;
8                  j ← k;
9                  mayor ← abs(matrix_{k,l});
10             end
11         end
12     end
13 end
```

Algorithm 3: Eigenpares Jacobi

```
1  eigen_jacobi (matrix, m, ERROR);
   Input  : matrix: es donde se guardan los coeficientes de la matriz,
            ERROR: tolerancia m: es la dimensión de la matriz
   Output: eigenvectores
2  i,j ← mayor_absoluto(matrix, m);
3  while abs(matrix_i, j) > ERROR do
4      numerador ← matrix_{i,j} * 2;
5      denominador ← matrix_{i,i} − matrix_{j,j};
6      if denominador < ERROR then
7          θ ← π/4;
8      else
9          θ ← atan(numerador/denominador) * 0.5;
10     end
11     s ← sin(θ);
12     c ← cos(θ);
13     s2 ← s * s;
14     c2 ← c * c;
15     cii = matrix_{i,i} * c2 + 2. * matrix_{i,j} * s * c + matrix_{j,j} * s2;
16     cjj = matrix_{i,i} * s2 − 2. * matrix_{i,j} * s * c + matrix_{j,j} * c2;
17     cij = matrix_{j,j} − matrix_{i,i} * s * c + matrix_{i,j} * (c2 − s2);
18     matrix_{i,j} = *(*(matrix + j) + i) = cij;
19     matrix_{j,j} = cjj;
20     matrix_{i,i} = cii;
21     for k ← 0 : m do
22         if k ≠ i and k ≠ j then
23             cik ← matrix_{i,k} * c + matrix_{j,k} * s;
                cjk ← −matrix_{i,k} * s + matrix_{j,k} * c;
24             matrix_{i,k} ← cik;
25             matrix_{k,i} ← cik; matrix_{j,k} ← cjk;
26             matrix_{k,j} ← cjk;
27         end
28     end
29     if iteration = 0 then
30         for l ← 0 : m do
31             for k ← 0 : m do
32                 if l = k then
33                     eigenvectores_{l,k} = 1
34                 end
35                 ;
36             end
37         end
38         eigenvectores_{i,i} ← c;
39         eigenvectores_{j,j} ← c;
40         eigenvectores_{i,j} = −s;
41         eigenvectores_{j,i} = s;
42     else
43         for l ← 0 : m do
44             eli ← eigenvectores_{l,i} * c + eigenvectores_{l,j} * s;
45             elj ← −eigenvectores_{l,i} * s + eigenvectores_{l,j} * c;
46             eigenvectores_{l,i} ← eli;
47             eigenvectores_{l,j} ← elj;
48         end
49     end
50     i, j ← mayor_absoluto(matrix, m);
51     iteration ← iteration + 1;
52 end
53 return eigenvectores;
```

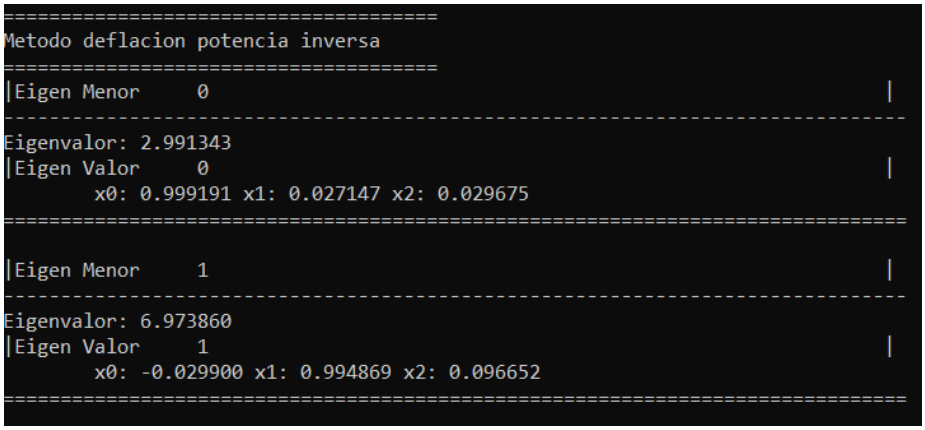
Resultados

En ambos algoritmos se tomo una tolerancia del 1E-13, en el algoritmo de deflación se compara el valor propio, y en el método de Jacobi se compara que los elementos de la matriz fuera de la diagonal tiendan a 0;

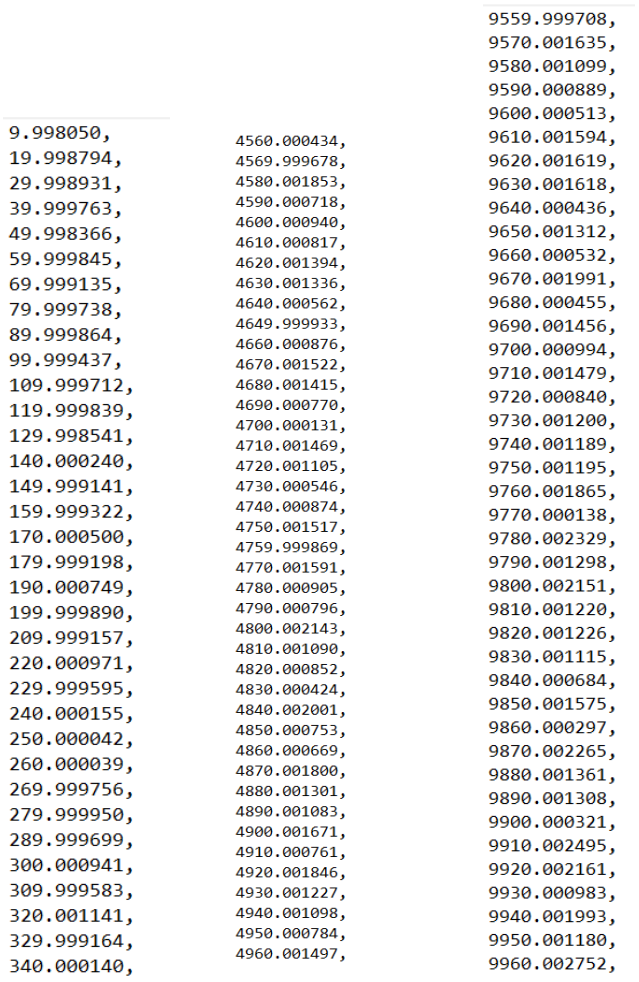
Deflación potencia Inversa

Todos los resultados se encuentran en la carpeta
"Resultados/deflacion_inversa/[].txt"

Figure 1: Matriz de 3x3



En seguida se muestran algunos de los eigenvalores obtenidos para las otras matrices que son de una dimensión mucho mayor.



(a) 50x50 (b) 500x500 (c) 1000x1000

Figure 2: a) Tiene los primeros eigenvalores, b) y c) los últimos

Jacobi

Todos los resultados se encuentran en la carpeta "Resultados/jacobi/[].txt"

Figure 3: Matriz de 3x3

```
=====================================
Metodo de Jacobi eigenpares
=====================================
El numero de iteraciones es: 13

Los eigenvalores en la diagonal son
-------------------------------------
10.0347962641139 0.0000000000000 -0.0000000000000
0.0000000000000 6.9738604132940 0.0000000000000
-0.0000000000000 0.0000000000000 2.9913433225921


Los eigenvectores en columnas son
-------------------------------------
-0.0268990178725 0.0298996912872 0.9991908983265
-0.0974605268897 -0.9948690894083 0.0271466505902
0.9948758156352 -0.0966514531748 0.0296750410835
-------------------------------------
Se han creado los archivos en Resultados/jacobi/_3x3_
```

En seguida se muestran algunos de los eigenvalores de las matrices de mayor dimensión.

419.999822	0.000218
410.000444	0.000217
400.000817	0.000211
390.000943	0.000204
380.001210	0.000196
369.999165	0.000191
360.001077	0.000189
349.999929	0.000182
340.000140	0.000182
329.999164	0.000173
320.001141	0.000173
309.999583	0.000171
300.000941	0.000164
289.999699	0.000156
279.999950	0.000142
269.999756	1.000000
260.000039	0.000141
250.000042	0.000135
240.000155	1.000000
229.999595	0.000133
220.000971	0.000132
209.999157	0.000127
199.999890	0.000113
190.000749	0.000104
179.999198	0.000101
170.000500	0.000098
159.999322	0.000086
149.999141	0.000084
140.000240	0.000074
129.998541	0.000072
119.999839	0.000068
109.999712	0.000061
99.999437	0.000048
89.999864	0.000046
79.999738	0.000038
69.999135	0.000036
59.999845	1.000000
49.998366	0.000026
39.999763	0.000018
29.998931	0.000017
19.998794	0.000008
9.998050	0.000002
(a) 50x50	(b) 125x125

Figure 4: a) Tiene los últimos eigenvalores

Conclusión

El método de la potencia inversa usada como deflación se vuelve demasiado tardado para matrices de tamaño alrededor de 500, en mi caso particular, utilizando 12GB de RAM y un procesador AMD Ryzen 5 de 3.2 GHZ, una matriz de 500x500 tardaba alrededor de 30 minutos para poder calcular sus 46 eigenvalores y eigenvectores, para la matriz de 1000 filas, tardo aproximadamente 5 horas en poder encontrar las soluciones que se pedían, entré más vectores queramos más se tardará debido a que se necesitan eliminar las contribuciones de

todos los eigenvectores anteriores, y en este lugar se hace un cuello de botella, además de que necesitamos utilizar un método para poder aproximar un eigenvector nuevo.

En contraste aunque el método de Jacobi no nos permite hacer operaciones con grandes dimensiones, esté me parece más fácil de implementar, además de que la solución converge rápidamente, también es importante que ya nos da como resultado los eigenvectores en una matriz.