Práctica 1 – Valores y vectores propios

fecha de entrega: 27 de Septiembre de 2017 hora 23:55

1. Implementación

Ejecuta el comando `format long` para mostrar más dígitos.

1.1. Método de la Potencia

Escribe una función encabezada:

```
function [sigmai, qi] = metodo_potencia(A, q0, k, tol)
```

en Matlab que aplique el método de la potencia a sus argumentos (una matriz `A`, un vector inicial `q0`, un número máximo `k` de iteraciones y una tolerancia `tol` para el error relativo). El resultado de la función debe ser la ultima de las parejas aproximativas (σ_i, q_i) al eigenpar dominante (λ_1, v_1).

1.2. Método de la Potencia inversa con shift

Escribe una función encabezada:

```
function [sigmai, qi] = metodo_potencia_inv(A, q0, rho, k, tol)
```

en Matlab que aplique el método de la potencia inversa con *shift* a sus argumentos (donde `A, q0, k, tol` son como arriba y `rho` es un *shift*). La función debe devolver σ_i (la aproximacion al eigenvalor más cercano a ρ en módulo) y el eigenvector asociado q_i.

1.3. Método de la Potencia inversa con cociente de Rayleigh

Escribe una función encabezada:

```
function [sigmai, qi] = metodo_potencia_invRayleigh(A, q0, k)
```

en Matlab que aplique el método de la potencia inversa con cociente de Rayleigh a sus argumentos (donde `A, q0, k` son como arriba). El resultado de la función debe ser una aproximación de un eigenpar (λ_j, v_j).

Consejo: En el caso que este método converge, el cociente de Rayleigh converge hacia un eigenvalor. Por lo tanto la matriz tiende a una matriz singular.

1.4. Método: QR simple y QR con *shifts* dinámicos

Sean `A, k, tol` como arriba. Escribe dos funciones encabezadas:

```
function [lambdas] = MQR_simple(A, k, tol)
```

y

```
function [lambdas] = MQR_dynamic(A, k, tol)
```

en MATLAB que apliquen el método QR simple y el método QR con *shift* dinámico. El resultado de cada de las funciones debe ser una aproximación de los eigenvalores $\lambda_j, (j = 1, \cdots, n)$ de la matriz A. Para terminar la iteración del método QR simple, use el criterio absoluto `norm(diag(Am, -1)) <tol`, (recuerda que $A_m \to T$ con T triangular superior). El criterio para el método QR con *shift* dinámico se encuentra en las notas de la clase, en este caso el parámetro `k` puede ser usado para limitar las iteraciones hacia un valor proprio.

Optativo: Para matrices simétricas pueden acumular Q_m para obtener los vectores proprios.

2. Verificación de la implementación y teoría

Ejercicio 1

Sea

$$A = \begin{pmatrix} 1 & 1 & 2 \\ -1 & 9 & 3 \\ 0 & -1 & 3 \end{pmatrix}.$$

1. Usando tu programa, aplica el método de la potencia a la matriz A, empezando con el vector inicial $q_0 = (1, 1, 1)$. Haz al menos 10 iteraciones.

2. Usa el comando `[V,D] = eig(A)` de MATLAB para obtener los valores y vectores propios "exactos" de A. Compara el valor propio dominante de `D` con tu resultado del apartado 1.

3. Sea v el vector propio dominante en `V` (dado por `eig` arriba). Ese está normalizado y tiene la norma euclidiana uno. Divide v por su elemento de mayor magnitud, con el fin de poder compararlo fácilmente con los vectores de tu iteración.

   Calcula las razones
   $$\frac{||q_{j+1} - v||_\infty}{||q_j - v||_\infty}, \qquad j = 1, 2, 3, \ldots.$$

4. Usando `D`, calcula la razón $|\lambda_2/\lambda_1|$ y compáralo con las razones del apartado 3.

Ejercicio 2

Repite el Ejercicio 1 con el método de la potencia inversa con *shift*:

- $\rho = 0$: Recuerde la razón exacta en este caso es $|\lambda_n/\lambda_{n-1}|$.
- $\rho = 3.5$: La razón exacta para ese caso se deja calcular usando `eig`.

Ejercicio 3

Usa la matriz del Ejercicio 1 y tu programa de potencia inversa con cociente de Rayleigh con varios vectores iniciales q_0 para calcular tres vectores propios linealmente independientes, y sus valores propios.

Además, verifica para alguno de los tres vectores q_0 y su límite (un eigenvector v_j) que la convergencia es cuadrática.

Ejercicio 4

Decide porqué el método de la potencia no funciona para la matriz siguiente:

$$A = \begin{pmatrix} 1 & -4 & -6 \\ -12 & -8 & -6 \\ 11 & 10 & 10 \end{pmatrix} .$$

Ejercicio 5

Compara los resultados de tu implementación del método QR simple y los del QR con *shift* dinámico con el resultado del comando MATLAB `eig`. Para ello usa matrices dados por el siguiente comando de MATLAB:

```
gallery( 'fiedler', 25 )
```

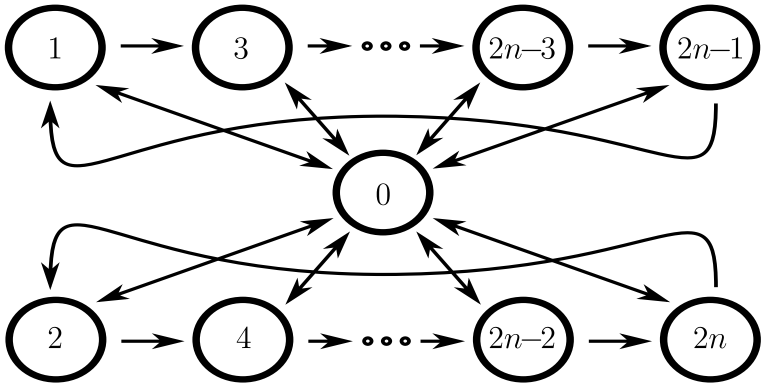
y aplica los métodos como sigue:

- `MQR_simple( gallery( 'fiedler', 25 ), 1000, 1E-10)`
- `MQR_dynamic( gallery( 'fiedler', 25 ), 20, 1E-10)`
- `eigs( gallery( 'fiedler', 25 ), 25)`

Además cuenta las iteraciones requeridas por tu implementación del método QR simple *vs* QR con *shift* dinámico.

Ejercicio 6

Para un n dado, considera los $2n+1$ nodos denotados por los números 0, 1, ..., $2n$. Construye la matriz de adyacencias si todos los nodos se conectan al nodo cero y los nodos pares se conectan de modo cíclico así como los nodos impares, es decir:



Conjetura con varios valores de n cuál es el comportamiento del *ranking* de Google del nodo 0 cuando $n \to \infty$. Repite el ejercicio cambiando todas las flechas unidireccionales por bidireccionales y conjetura de nuevo.