UNIVERSIDAD DE
MURCIA

DEPARTAMENTO DE MATEMÁTICAS
ANTONIO J. PALLARÉS

ANÁLISIS NUMÉRICO MATRICIAL CURSO 2019-2020

PRIMERA ENTREGA PRACTICA
Entrega antes del día 25/04/2020

La entrega práctica consta de dos ejercicios y se valorará sobre 10 puntos (un 25 % de la nota de prácticas).

Crea en tu proyecto una carpeta `anm1entrega`, donde ir poniendo los ficheros correspondientes a esta práctica. Utiliza scripts `ejercicioxx.m` para cada ejercicio (xx es el número del mismo). Incluye con comentarios en cada fichero tu análisis de los resultados que producen.

Ejercicio 5.1

En prácticas hicimos el ejercicio 4.3 de la práctica 4, en el que modificamos el código de la función LUPQGauss.m para hacer la función LUPQGeneral.m

[P,L,U,Q,r]=PLUQfGeneral(A)

que devuelve: el rango r de la matriz A, una matriz permutación (filas) P de dimensión m, una matriz triangular inferior con unos en la diagonal L de m filas y min(m,n) columnas, una matriz tringular superior U de min(m,n) filas y n columnas, y una matriz permutación (columnas) Q de dimensión n, tales que

$$LU = PAQ \Longleftrightarrow P'LU = AQ.$$

El próposito de este ejercicio es hacer una función nucleo.m

N=nucleo(A)

que devuelva una matriz N *cuyas n-r columnas que constituyen una base del núcleo de la aplicación lineal* $x \to Ax$ *definida en* $\mathbb{R}^n$ *con valores en* $\mathbb{R}^m$*, para* A *de rango* r*. Para ello os propongo seguir el siguiente procedimiento:*

1. *Utiliza matrices aleatorias de m filas, n columnas y rango r (para distintos valores) construida como en el ejercicio 4.4 para ir comprobando las afirmaciones:*

   a) *Como U proviene del método de eliminación de Gauss Total, interrumpido cuando sólo quedan ceros para las filas entre r+1 y m, su forma es*

   $$U = \left[\begin{array}{c|c} Ur & G \\ \hline 0 & 0 \end{array}\right]$$

   donde Ur=U(1:r,1:r) tiene dimensión $r \times r$ *y G=U(1:r,r+1:n) tiene dimensión* $r \times (n-r)$.

   b) *Los vectores columna de la matriz K formada por la solución Kr del sistema tridiagonal* $Ur * Kr = -G$ *a la que se le añaden las filas de la matriz identidad de dimensión (n-r):*

   $$K = \left[\begin{array}{c} Kr \\ \hline eye(n-r) \end{array}\right]$$

   forman una base del núcleo de U. En consecuencia, los vectores columna de $Q * K$ *forman una base del núcleo de A.*

2. *Reune el proceso de los apartados anteriores para crear una función nucleo.m*

   Utiliza las mismas matrices del apartado anterior para comprobar su funcionamiento.

Ejercicio 5.2

Incluye en el directorio Matrices *los métodos iterativos de valores y vectores propios que encontraras en* `ParaEntrega1.zip`*:*

potencia.m

potenciaInvDesplazada.m

Comprueba su funcionamiento con el ejemplo manejado en clase *ejemploPotenciaValoresVectoresPropios.m*

1. *Construye la matriz simétrica "test" A definida por*

   d=[2,3,1,5];
   D= diag(d);

   ```
   U=[ 1,-1, 0, 1;
   1, 1,-1, 0;
   0, 1, 1, 1;
   1, 0, 1,-1];
   ```

   *U= 1/sqrt(3)*U;*
   *A= U'*D*U*

2. *Comprueba el funcionamiento de los métodos de la potencia y la potencia inversa para aproximar los valores propios de mayor y menor tamaño de la matriz "test" del apartado anterior.*

   Cambia los valores del vector d *que contiene el espectro de* A *para ver como funcionan.*

3. *Sabiendo que los valores propios de la matriz*

$$A = \begin{pmatrix} -2 & 0 & 1 & 1 \\ -5 & -5 & 4 & 5 \\ 16 & 14 & -8 & -12 \\ -19 & -15 & 11 & 15 \end{pmatrix}$$

   son de la forma $\lambda_4 = -\lambda_1 < \lambda_3 = -\lambda_2 < 0 < \lambda_2 < \lambda_1$*, encontrar una base de* $\mathbb{R}^4$ *formada por vectores propios de* A*.*

Ejercicio 5.3

En el fichero QR-Burden-Faires.pdf *está la sección del libro de Burden y Faires dedicado al algoritmo de factorización QR para matrices tridiagonales simétricas y su uso para la aproximación de valores propios*

El propósito de este ejercicio es que implementéis el primer paso del proceso, obteniendo la factorización QR de una matriz tridiagonal simétrica $T = QR$*, donde la matriz ortogonal* Q *se obtiene como producto de rotaciones (giros)*

$$P_{i+1} = \begin{array}{c} \begin{array}{cccccccc} & & & i & i+1 & & & \\ & & & \downarrow & \downarrow & & & \end{array} \\ \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \cos\theta_i & \operatorname{sen}\theta_i & & & \\ & & & -\operatorname{sen}\theta_i & \cos\theta_i & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \end{array} \begin{array}{l} \\ \\ \\ \\ \leftarrow \quad i \\ \leftarrow \quad i+1 \\ \\ \\ \\ \end{array} \tag{1}$$

$Q' = P_nP_{n-1}...P_2$. $R = A_n$ se obtiene en n-1 etapas, A_k, empezando con $A_1 = T$, anulando en cada etapa el elemento no nulo que hay bajo la diagonal en la columna k mediante una rotación: $A_{k+1} = P_{k+1}A_k$.

Siguiendo el primer apartado del pdf adjunto, construye una función que devuelva la factorización QR de T:

[Q,R]=QRTRid(T)

Ind: Para utilizar el mínimo espacio puedes hacer una función intermedia

[c,s,z,q,r]=QRTridVec(dp,di)

que admita como datos de entrada los vectores dp *y* di*, con las diagonales principal e inferior (coincide con la superior) de la matriz tridiagonal T, y devuelva los vectores* c *y* s *con los cosenos y senos correspondientes a las rotaciones P_i, la diagonal principal* z *de R y las dos diagonales superiores* q *y* r *de R.*

Comprueba su funcionamiento con matrices tridiagonales simétricas de dimensión n=3,6 y 8, construidas de manera aleatoria.