Tarea 7

Fecha de publicación: Octubre 8, 2020

Fecha de entrega: Domingo 25 de octubre de 2020.

Ejercicio 1. (2 puntos)
Sea $\boldsymbol{V} = [\boldsymbol{v}_1 \ ... \ \boldsymbol{v}_n]$ una matriz de tamaño n, donde $\boldsymbol{v}_l$ es su columna l-ésima. Dados un ángulo θ y los índices i y j, se define la rotación de Givens $\boldsymbol{G}(i, j, \theta)$. El objetivo es calcular el producto $\boldsymbol{V}\boldsymbol{G}(i, j, \theta)$ sin tener que construir la rotación de Givens.

1. Si $\boldsymbol{P} = [\boldsymbol{p}_1 \ ... \ \boldsymbol{p}_n] = \boldsymbol{V}\boldsymbol{G}(i, j, \theta)$ es la matriz producto, muestre que la columna $\boldsymbol{p}_l$ es igual a $\boldsymbol{v}_l$ si $l \neq i, j$, y escriba las expresiones en el reporte para calcular $\boldsymbol{p}_i$ y $\boldsymbol{p}_j$ en términos de las columnas de $\boldsymbol{V}$ y los valores del seno y coseno del ángulo θ.

2. Escriba una función que calcula el producto $\boldsymbol{V}\boldsymbol{G}(i, j, \theta)$. Esta función recibe como argumentos el apuntador a una matriz $\boldsymbol{V}$, su tamaño n, los índices i y j, y los valores $c = \cos\theta, s = \sin\theta$ que describen a la rotación de Givens $\boldsymbol{G}(i, j, \theta)$.
   La función debe sobreescribir las entradas de la matriz $\boldsymbol{V}$ para reemplazarlas por el resultado del producto $\boldsymbol{V}\boldsymbol{G}(i, j, \theta)$. Esto es para no tener que crear una nueva matriz que corresponda al producto.

   Note que necesita hacer una copia de las columnas $\boldsymbol{v}_i$ y $\boldsymbol{v}_j$ antes de sobreescribir los valores.

Ejercicio 2. (2 puntos)

1. Escriba una función que calcule el producto $\boldsymbol{V}\boldsymbol{D}$ donde $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, ..., d_n)$ sin tener que construir la matriz diagonal. Esta función recibe como argumentos el apuntador a una matriz $\boldsymbol{V}$, su tamaño n, y el apuntador a un arreglo $\boldsymbol{d} = (d_1, d_2, ..., d_n)$.
   Calcule las entradas de la matriz producto $\boldsymbol{V}\boldsymbol{D}$ usando los valores $d_1, ..., d_n$, y que la función devuelva la matriz producto.

Ejercicio 3. (6 puntos)
Programar el método iterativo de Jacobi para calcular los eigenvalores y eigenvectores de una matriz simétrica.

1. Escriba la función que corresponde al Algoritmo 1 que se muestra más adelante.

   En el algoritmo se mencionan los parámetros de entrada de la función y cuáles serán su salida, que pueden ser codificados como variables que se pasan por referencia como argumentos a la función o definir una estructura, etc. Es opcional la forma de hacerlo.

   Note que el algoritmo está descrito para no tener que construir la rotación de Givens y que se vaya actualizando la matriz $\boldsymbol{B}$ que corresponde a la matriz $\boldsymbol{A}^{(k-1)}$ del algoritmo visto en la Clase 13, para no tener que crear estas matrices en cada iteración. Por esta razón es que en el paso (3.7) se requiere almacenar una copia de los vectores $\boldsymbol{b}_i$ y $\boldsymbol{b}_j$ que corresponden a las columnas i y j de la matriz $\boldsymbol{B}$ antes de sobreescribir sus valores en los pasos siguientes. Se denota por $(\boldsymbol{b}_i)_l$ a la componente l-ésima del vector $\boldsymbol{b}_i$.

   En el punto (3.9) hay que usar el Ejercicio 1 para actualizar las entradas de la matriz $\boldsymbol{V}$ sin tener que calcular la rotación de Givens.

   También note que como la matriz $\boldsymbol{B}$ es simétrica, en el punto (3.1) del algoritmo, sólo hay que buscar en los elementos que están por arriba de la diagonal o por abajo de ella para encontrar el elemento más grande en valor absoluto.

2. Escribir un programa que muestre el resultado del método iterativo de Jacobi. El programa debe recibir desde la línea de comandos el nombre de un archivo que contiene a los elementos de una matriz simétrica.

   Haga que el programa lea el archivo de datos, cree la matriz correspondiente e imprima el tamaño n de la matriz.

   Defina la tolerancia τ como $\sqrt{\epsilon_m}$ y el número máximo de iteraciones como $M = 100n^2$.

   Usando la función del Algoritmo 1, obtenga la matriz de eigenvectores $\boldsymbol{V}$, el vector de eigenvalores $\boldsymbol{d}$, $b_{\max}$ y el número k de iteraciones realizadas.

   Haga que el programa imprima los valores k, $b_{\max}$ y los primeros y últimos elementos del vector $\boldsymbol{d}$.

   Usando la función del Ejercicio 2, obtenga el producto $\boldsymbol{VD}$, donde $\boldsymbol{D}$ es la matriz diagonal que tiene por elementos los eigenvalores $\boldsymbol{d}$, calcule el error $\|\boldsymbol{AV} - \boldsymbol{VD}\|$, e imprimalo en la consola.

3. Use el archivo `datosTarea07.zip` que contiene los archivos de tres matrices simétricas. Pruebe el programa anterior con esos datos y agregue un comentario en el reporte si se logró diagonalizar la matriz $\boldsymbol{A}$ aplicando las rotaciones de Givens, sobre el error

en la descomposición espectral de la matriz y la cantidad de iteraciones que hay que realizar comparando con el tamaño de la matriz.

Algoritmo 1: Método iterativo de Jacobi.

Data: Dar una matriz simétrica $\boldsymbol{A}$, su tamaño n, una tolerancia $\tau > 0$ y un número M de iteraciones máximas. Inicializar $\boldsymbol{V} = \boldsymbol{I}$.

Result: La matriz $\boldsymbol{V}$ con los eigenvectores de la matriz,
el arreglo $\boldsymbol{d}$ con los eigenvalores correspondientes,
el número k de iteraciones realizadas y
el valor del elemento $b_{\max}$, que es el más grande en valor absoluto.

(1) Inicializar la matriz $\boldsymbol{B} = [b_{ij}]$ como una copia de la matriz $\boldsymbol{A}$;
(2) Inicializar la matriz $\boldsymbol{V}$ como la matriz identidad $\boldsymbol{I}$ de tamaño n;
for $k = 1, 2, ..., M$ **do**
 (3.1) Encontrar los índices i, j del elemento $b_{\max} = b_{ij}$ de la matriz $\boldsymbol{B}$ más grande, en valor absoluto, fuera de su diagonal;
 (3.2) Terminar de iterar si $|b_{\max}| < \tau$;
 (3.3) Calcular $\delta = (b_{jj} - b_{ii}) \,/\, (2b_{ij})$;
 (3.4) $t = sign(\delta)/(|\delta| + \sqrt{1+\delta^2})$;
 (3.5) $c = \cos\theta = 1/\sqrt{1+t^2}$;
 (3.6) $s = \sin\theta = ct$;
 (3.7) Definir los vectores $\boldsymbol{b}_i$ y $\boldsymbol{b}_j$ como las columnas i y j de $\boldsymbol{B}$;
 (3.8) Actualizar la matriz $\boldsymbol{B} = [B_{rt}]$ para que coincida con $\boldsymbol{G}^\top \boldsymbol{B}\boldsymbol{G}$:
 (3.8.1) Crear las copias $\boldsymbol{b}_i$ y $\boldsymbol{b}_j$ de las columnas i, j de $\boldsymbol{B}$;
 (3.8.2) $B_{ii} = c^2 \cdot (\boldsymbol{b}_i)_i - 2sc \cdot (\boldsymbol{b}_i)_j + s^2 \cdot (\boldsymbol{b}_j)_j$;
 (3.8.3) $B_{jj} = s^2 \cdot (\boldsymbol{b}_i)_i + 2sc \cdot (\boldsymbol{b}_i)_j + c^2 \cdot (\boldsymbol{b}_j)_j$;
 for $l = 1, 2, ..., n,\ \ l \neq i, j$ **do**
  (3.8.4.1) $B_{il} = c \cdot (\boldsymbol{b}_i)_l - s \cdot (\boldsymbol{b}_j)_l$;
  (3.8.4.2) $B_{jl} = s \cdot (\boldsymbol{b}_i)_l + c \cdot (\boldsymbol{b}_j)_l$;
  (3.8.4.3) $B_{li} = B_{il}$;
  (3.8.4.4) $B_{lj} = B_{jl}$;
 end

 (3.8.5) Hacer $B_{ij} = B_{ji} = 0$;
 (3.9) Calcular $\boldsymbol{V} = \boldsymbol{V}\boldsymbol{G}$ (usando las operaciones del ejercicio 1);
end

(4) Definir el vector $\boldsymbol{d}$ como la diagonal de la matriz $\boldsymbol{B}$;
