Tarea 4

Fecha de publicación: Septiembre 17, 2020

Fecha de entrega: Domingo 27 de septiembre de 2020.

Ejercicio 1 (5 puntos).
Programar el algoritmo de factorización LU con pivoteo parcial y probarlo resolviendo un sistema de ecuaciones lineales.

1. Programe la función `factLU` que calcula la factorización LU de acuerdo al Algoritmo 1, y que reciba como parámetros:

   - el apuntador a la matriz $\boldsymbol{A}$,
   - el tamaño n de la matriz,
   - una tolerancia τ,
   - el apuntador a una matriz $\boldsymbol{L}$ de tamaño $n \times n$,
   - el apuntador a una matriz $\boldsymbol{U}$ de tamaño $n \times n$,
   - el apuntador a un vector $\boldsymbol{p}$ de tamaño n.

   La función debe devolver 0 si logró hacer la factorización y 1 si no lo hizo.
   Vamos a considerar que previamente se reservó memoria para $\boldsymbol{L}$, $\boldsymbol{U}$ y $\boldsymbol{p}$, de modo que la función debe cambiar los valores de las entradas de estos arreglos.
   Hay que tener cuidado al escribir el código porque el algoritmo está descrito de modo que los índices de las matrices y vectores empiezan en 1, mientras que en C o Python empiezan en 0.

2. Escriba la función `genSolLU` que resuelve el sistema $\boldsymbol{LUx} = \boldsymbol{Pb}$, donde $\boldsymbol{L}$ es una matriz triangular inferior y $\boldsymbol{U}$ es una matriz triangular superior.
   La función debe recibir como parámetros:

   - el apuntador a la matriz $\boldsymbol{L}$,
   - el apuntador a la matriz $\boldsymbol{U}$,
   - el tamaño n de la matriz,
   - el vector $\boldsymbol{b}$,

- el apuntador a un arreglo de enteros $\boldsymbol{p}$ de longitud n,
- una tolerancia τ.

Crear un arreglo $\hat{\mathbf{b}} = (\hat{b}_1, ..., \hat{b}_n)^\top$ con los elementos de $\boldsymbol{b} = (b_1, ..., b_n)^\top$ reordenados de acuerdo a $\boldsymbol{p} = (p_1, ..., p_n)^\top$, esto es, $\hat{b}_i = b_{p_i}$.

Use las funciones `backwardSubstitution` y `forwardSubstitution` de la tarea 3 para resolver el sistema de ecuaciones. Si no hay ningón problema, la función debe devolver el apuntador al vector solución $\boldsymbol{x}$. En caso contrario, devolver `NULL`.

3. Escriba la función `solFactLU` que resuelva el sistema de ecuaciones $\boldsymbol{Ax} = \boldsymbol{b}$ usando la factorización LU. La función debe hacer lo siguiente:

   (a) Crear las matrices $\boldsymbol{L}$, $\boldsymbol{U}$ y un arreglo unidimensional $\boldsymbol{p}$ de tamaño n.

   (b) Usar la función `factLU` para obtener la factorización LU de la matriz $\boldsymbol{A}$.

   (c) Si no se logró factorizar la matriz, libere la memoria y devuelva `NULL`.

   (d) En caso contrario, use la función `genSolLU` del punto anterior para resolver el sistema de ecuaciones $\boldsymbol{LUx} = \boldsymbol{Pb}$, libere memoria y devuelva el apuntador a $\boldsymbol{x}$ o `NULL` según el resultado de la función.

   Note que como la factorización es $\boldsymbol{PA} = \boldsymbol{LU}$, entonces para resolver $\boldsymbol{Ax} = \boldsymbol{b}$, habría que multiplicar por $\boldsymbol{P}$ ambos miembros: $\boldsymbol{PAx} = \boldsymbol{Pb}$, pero esto es lo mismo que $\boldsymbol{LUx} = \boldsymbol{Pb}$, y $\boldsymbol{Pb}$ se obtiene cambiando de orden los elementos de $\boldsymbol{b}$ según las entradas de $\boldsymbol{p}$.

4. Escriba un programa que lea desde la línea de comandos los nombres de dos archivos binarios, uno que tiene la información de una matriz $\boldsymbol{A}$ y el otro con la información del vector de términos independientes $\boldsymbol{b}$.
   Haga que el programa lea la información de los archivos y cree la matriz $\boldsymbol{A}$ y el vector $\boldsymbol{b}$ y que imprima el tamaño de la matriz y el tamaño del vector $\boldsymbol{b}$.
   Use la función del inciso anterior para calcular la solución de $\boldsymbol{Ax} = \boldsymbol{b}$. Si la función devuelve `NULL`, imprima el mensaje de que la matriz es singular.
   Si el sistema tuvo solución, use la función de la tarea 3 para imprimir las primeras y últimas entradas del vector solución. Finalmente imprima el valor del error $\|\boldsymbol{Ax}-\boldsymbol{b}\|$ y libere la memoria.

5. Pruebe el programa usando los datos del archivo `datosLU.zip`, usando las parejas de archivos:
   ```
   matrizA5     vectorb5
   matrizA50    vectorb50
   matrizA500   vectorb500
   ```
   Copie las salidas del programa al archivo de respuestas.

Algoritmo 1: Factorización LU con pivoteo parcial.

Data: La matriz $\boldsymbol{A}$, su tamaño n y una tolerancia τ. Adicionalmente se pasan unas matrices $\boldsymbol{L}$ y $\boldsymbol{U}$, y un arreglo $\boldsymbol{p}$ de tamaño n para que sus entradas sean modificadas por el algoritmo.

Result: Las matrices $\boldsymbol{L}$ y $\boldsymbol{U}$ de la factorización LU,
un vector $\boldsymbol{p}$ que indica la permutación $\boldsymbol{P}$ tal que $\boldsymbol{PA} = \boldsymbol{LU}$,
y devolver 0 si el algoritmo concluyó exitósamente o 1 si no lo hizo.

— *Inicialización* —

Inicializar $\boldsymbol{L} = [l_{ij}]$ como la matriz identidad de tamaño n;
Copiar a $\boldsymbol{U} = [u_{ij}]$ las entradas de la matriz $\boldsymbol{A}$;
Inicializar $\boldsymbol{p} =$ p_{0}, ... , p_{n-1} de modo que $p_i = i$;

for k = 0,1, ... , n-1 **do**
    /* Encuentra la ecuación pivote */
    Encontrar el índice r tal que $|u_{rk}| = \max_{k \le i \le n} |u_{ik}|$;
    if $|u_{rk}| < \tau$ **then**
        Terminar devolviendo el valor 1;
    end
    if $r \neq k$ **then**
        /* Intercambia las filas k y r */
        for j = 0, 1, ... , n-1 **do**
            $v = u_{kj}$;
            $u_{kj} = u_{rj}$;
            $u_{rj} = v$;
        end
        /* Registra el cambio de filas en el arreglo de la permutación */
        $v = p_k$;
        $p_k = p_r$;
        $p_r = v$;
        if k > 0 **then**
            /* Realiza el cambio de filas en la matriz L */
            for j j = 0, 1, ... , k-2 $k-1$ **do**
                $v = l_{kj}$;
                $l_{kj} = l_{rj}$;
                $l_{rj} = v$;
            end
        end
    end
    for i i = k, k+1, ... , n-1 $.., n$ **do**
        $l_{ik} = u_{ik}/u_{kk}$;
        for j = k-1, k, ... , n-1 n **do**
            $u_{ij} = u_{ij} - l_{ik}u_{kj}$;
        end
    end
end
Terminar devolviendo el valor 0;

`Nota:` En el bloque de inicialización del algoritmo se indica que se copien los elementos de $\boldsymbol{A}$ a la matriz $\boldsymbol{U}$. Como todas las operaciones se hacen sobre la matriz $\boldsymbol{U}$, la matriz $\boldsymbol{A}$ queda intacta.

Ejercicio 2 (5 puntos).
Programar el algoritmo de factorización de Cholesky y resuelva un sistema de ecuaciones lineales.

1. Escriba la función `factChol` calcule la matriz $\boldsymbol{L}$ de la factorización de Cholesky. Esta función recibe como parámetros:

   - el apuntador a una matriz $\boldsymbol{A}$,
   - su tamaño n,
   - una tolerancia τ.

   Dentro de la función se reserva memoria para la matriz $\boldsymbol{L}$ y se inicializa como la matriz cero.
   Si al cálcular l_{jj} el radicando es negativo o $|l_{jj}|$ es menor que la tolerancia τ, liberar la memoria de $\boldsymbol{L}$ y devolver `NULL`. Si no ocurre lo anterior, devuelve el apuntador a $\boldsymbol{L}$.

2. Escriba la función `transpose` que recibe como argumentos el apuntador a una matriz $\boldsymbol{M}$ y su tamaño n. La función crea matriz y llena sus entradas con los elementos de la transpuesta de $\boldsymbol{M}$. La función devuelve el apuntador a la matriz creada.

3. Escriba la función que resuelve el sistema $\boldsymbol{Ax} = \boldsymbol{b}$ cuando $\boldsymbol{A}$ es una simétrica y definida positiva. La función debe recibir como parámetros:

   - el apuntador a una matriz $\boldsymbol{A}$,
   - su tamaño n,
   - el apuntador al vector $\boldsymbol{b}$,
   - una tolerancia τ,

   La salida de la función es el apuntador al vector $\boldsymbol{x}$ o `NULL` si no se logró factorizar la matriz.

   Use la función del `factChol` primer punto para calcular la factorización de Cholesky de $\boldsymbol{A}$. Si el resultado es `NULL`, terminar devolviendo `NULL`.
   En caso contrario, use la función `transpose` para obtener $\boldsymbol{L}^\top$.
   Use la `genSolLU` del ejercicio 1 para resolver el sistema $\boldsymbol{LL}^\top\boldsymbol{x} = \boldsymbol{b}$, inicializando el vector $\boldsymbol{p}$ con la secuencia $0, 1, ..., n-1$.
   Liberar memoria y devolver el apuntador de $\boldsymbol{x}$ o `NULL` según el caso.

4. Escriba un programa que lea desde la línea de comandos los nombres de dos archivos binarios, uno que tiene la información de una matriz $\boldsymbol{A}$ y el otro con la información del vector de términos independientes $\boldsymbol{b}$.
   Haga que el programa lea la información de los archivos y cree la matriz $\boldsymbol{A}$ y el vector $\boldsymbol{b}$ y que imprima el tamaño de la matriz y el tamaño del vector $\boldsymbol{b}$.
   Use la función del punto anterior para resolver el sistema $\boldsymbol{Ax} = \boldsymbol{b}$. Dependiendo del resultado, imprima el mensaje de si no pudo calcular la solución o, en caso contrario, use la función de la tarea 3 para imprimir las primeras y últimas entradas del vector $\boldsymbol{x}$ e imprima el error $\|\boldsymbol{Ax} - \boldsymbol{b}\|$.

5. Pruebe el programa usando los datos del archivo `datosChol.zip`, usando las parejas de archivos:

`matrizSim5`	`vectorb5`
`matrizSim50`	`vectorb50`
`matrizSim500`	`vectorb500`

   Copie las salidas del programa al archivo de respuestas.