Tarea 7 - Métodos numéricos

Giovanni Gamaliel López Padilla

Índice

1. Introducción

La factorización de una matriz es usada para reducir la complejidad de la solución a un sistema de ecuaciones lineales. La factorización de una matriz da como resultado la multiplicación de dos matrices, una del tipo triangular inferior y la otra del tipo triangular superior. Esto es:

$$A = LU \tag{1}$$

donde L es la matriz triangular inferior y U matriz triangular superior.

2. Métodos

2.1. Factorización LU - Crout

Si A es una matriz de tamaño n×n, entonces su factorización en matrices L y U deben seguir la forma de las ecuaciones 2 y 3.

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \tag{2}$$

$$U = \begin{pmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{3}$$

Para calcular los elementos de cada matriz se tiene el conjunto de ecuaciones 4.

$$\begin{cases} l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj} \\ u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}}{l_{ii}} \end{cases} \tag{4}$$

El algoritmo planteado para la solución de este problema se encuentran en las carpetas Problema_1 y Problema_2, en el archivo LU_decomposition.h en la función obtain_LU_crout.
Al tener los elementos determiandos de las matrices L y U, es más sencillo realizar la solución a un sistema de ecuacuiones, ya que, Al tratarse de matrices triangulares sus algoritmos son más faciles de implementar. Suponiendo que se tiene el sistema $Ax = B$, entonces:

$$\begin{aligned} Ax &= B \\ LUx &= B \end{aligned}$$

nombrando

$$Ux = y \tag{5}$$

, entonces:

$$Ly = B \tag{6}$$

donde las ecuaciones 5 y 6 son dos sistemas de ecuaciones lineales del tipo triangular.
El algoritmo planteado para realizar la factorización LU con la version de Crount es la siguiente:

```
    // input: matriz
    // output: L,U
    for (int i = 0; i < n; i++)
    {
        for (int j = i; j < n; j++)
        {
            sum_ij = 0;
            for (int k = 0; k <= i - 1; k++)
            {
                sum_ij += l_ik * u_kj;
            }
            l_ij = matrix_ij - sum_ij;
        }
        for (int j = i + 1; j < n; j++)
        {
            sum_ij = 0;
            for (int k = 0; k < j - 1; k++)
            {
```

```
                    sum_ij += l_ik * u_kj;
                }
                u_ij = (matrix_ij - sum_ij) / l_ii;
            }
        }
```

Al termino de este de implementaron las funciones solve_triangular_inferior_matrix y solve_triangular_superior_matrix.

2.2. Factorización Cholesky

Sea A, una matriz simetica y definida positiva puede ser factorizada en la forma LU con $U = L^T$, esto es descrito en la ecuacuión 7.

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

El algoritmo emplea el conjunto de ecuaciones .

$$\begin{cases} l_{ii} = \sqrt{a_{ii} - \sum\limits_{k=1}^{i-1} l_{ik}^2} \\ l_{ij} = \frac{a_{ij} - \sum\limits_{k=1}^{j-1} l_{ik} l_{kj}}{l_{jj}} \end{cases} \tag{8}$$

El algoritmo de factorización de Cholesky se encuentra en la carpetas Problema_3_test y Problema_3_1 en el archivo Cholesky_decomposition.h en la función obtain_Cholesky. La función es la siguiente:

```
    // inputs: matrix, n
    // output
    for (int i = 0; i < n; i++)
    {
        sum = 0;
        for (int j = 0; j < i; j++)
        {
            sum += l_ik * l_ik;
        }
        sqrt_number = matrix_ii - sum;
        vadidate_positive_matrix(sqrt_number);
        validate_l_ii(l_ii);
        for (int j = i; j < n; j++)
        {
            sum = 0;
            for (int k = 0; k < i; k++)
            {
                sum += l_ik * l_jk;
            }
            l_ij = (matrix_ij - sum) / l_ii;
        }
    }
    fill_L_transpose(l, lt);
```

2.3. Factorización Doolittle

El método de factorización se basa en la eliminación Gaussiana. A partir de una matriz de tamaño n×n se obtiene una matriz triangular superior realizando operaciones entre columnas y renglones de la misma matriz. Entonces, la factorización de Doolittle se concentra en encontrar la matriz triangular inferior tal que se cumple la ecuación 9.

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ l_{31} & l_{32} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdot & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ 0 & 0 & \cdots & u_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdot & u_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdot & a_{nn} \end{pmatrix} \tag{9}$$

Realizando la multiplicación de matrices se obtiene que:

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ l_{21} & l_{21}u_{12} + u_{22} & \cdots & l_{21}u_{1n} + u_{2n} \\ l_{31} & l_{31}u_{12} + l_{32}u_{22} & \cdots & l_{31}u_{1n} + l_{3n}u_{22} + u_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n1}u_{12} + l_{n2} + u_{22} & \cdots & l_{n1}u_{1n} + l_{n2}u_{2n} + \cdots + u_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdot & a_{nn} \end{pmatrix} \tag{10}$$

Observando la diagonal de la ecuación 10, se puede obtener la ecuación 11.

$$u_{ii} = a_{ii} - \sum_{j=1}^{i} l_{ij}u_{ji} \tag{11}$$

Los elementos en los queda se obtiene un elemento de U sin realizar una multiplicación con un termino de L se obtiene la ecuación 12.

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{ki} \tag{12}$$

De los elementos donde un elemento de L esta multiplicando a un termino de la diagonal de U, se obtiene la ecuación 13.

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}}{u_{jj}} \tag{13}$$

3. Resultados

3.1. Factorización LU - Crout

3.1.1. Matriz de prueba

Se realizaron diferentes pruebas para el algoritmo de factorización LU usando la versión de Crout. La primera fue realizada con la matriz de la ecuación 14.

$$A = \begin{pmatrix} 3 & -1 & 4 & -1 \\ -1 & -1 & 3 & 1 \\ 2 & 3 & -1 & -1 \\ 7 & 1 & 1 & 2 \end{pmatrix} \qquad B = \begin{pmatrix} 10 \\ 5 \\ 1 \\ -20 \end{pmatrix} \tag{14}$$

Se obtuvo la factorización es la siguiente:

$$A = \begin{pmatrix} 3 & 0 & 0 & 0 \\ -1 & -1.34 & 0 & 0 \\ 2 & 3.67 & 8.25 & 0 \\ 7 & 3.34 & 2.5 & 5.54 \end{pmatrix} \begin{pmatrix} 1 & -0.34 & 1.34 & -0.34 \\ 0 & 1 & -3.25 & -0.5 \\ 0 & 0 & 1 & 0.18 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{15}$$

Y la solución del sistema de ecuaciones asociado es el siguiente:

$$\begin{cases} x_1 & = -2 \\ x_2 & = 1 \\ x_3 & = 3 \\ x_4 & = -5 \end{cases} \tag{16}$$

El sistema de la ecuación 14 fue resuelto en anteriores tareas y se obtuvieron los mismo resultados expuestos en las ecuaciones 16.

3.1.2. Archivos LARGE.txt

Usando los archivos M_LARGE.txt y V_LARGE.txt se obtuvieron los resultados contenidos en el archivo Solution_Large.txt, el cual se encuentra en la carpeta Problema_1.

3.1.3. Archivos SMALL.txt

El sistema de ecuaciones mostrado en la ecuación 17 esta contenida en los archivos M_SMALL.txt y V_SMALL.txt.

$$A = \begin{pmatrix} 2.402822 & 4.425232 & 1.929374 & 1.370355 \\ 1.201411 & 2.212616 & 0.964687 & 0.685178 \\ 1.119958 & 0.964687 & 2.053172 & 0.566574 \\ 0.742142 & 0.685178 & 0.566574 & 1.696828 \end{pmatrix} \qquad B = \begin{pmatrix} 0.060000 \\ 0.542716 \\ 0.857204 \\ 0.761270 \end{pmatrix} \tag{17}$$

Al introducir la matriz a el proceso de factorización LU con la versión de Crout da error, ya que un termino de la diagonal de la matriz L es igual a cero, por ende provocando una indeterminación. Por lo tanto, no es posible resolver el sistema de ecuaciones 17 por medio de la factorización LU. El programa con la información de la matriz se encuentra en la carpeta Problema_2

3.2. Factorización Cholesky

3.2.1. Matriz de prueba

Se probo el metodo de factorización de Cholesky con la matriz de la ecuación 18.

$$A = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 5 & -5 \\ 1 & -5 & 6 \end{pmatrix} \tag{18}$$

El programa que ejecuta este proceso es el contenido en la carpeta Problema_3_test, el resultado que se obtuvo es la ecuación 19, el cual se encuentra en los archivos L_test.txt y LT_test.txt.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{pmatrix} \tag{19}$$

3.2.2. Matrices dadas

Se factorizo las matrices de tamaño n×n, donde $n \in \{4, 50, 100\}$, donde los elementos de la matriz son los descritos en la ecuación

$$a_{ij} = \begin{cases} 2, & \text{si } i = j \\ -1, & \text{si } |i - j| = 1 \\ 0, & \text{otro} \end{cases} \tag{20}$$

La cual es generada en la función create_matrix que se encuentra en el archivo functions.h en la carpeta Problema_3_1. Los resultados para los diferentes valores de n se encuentran en los archivos L_n.txt y LT_n.txt en la carpeta Problema_3_1.

4. Conclusiones

Los métoodos de factorización de matrices ayudan a reducir el nivel de complejidad un sistema de matrices, al transformarlo a la solución de dos martrices triangulares, el defecto que pueden tener es no ser un método el cual sea apto a todo tipo de matrices. Esto puede verse en el caso del problema propuesto en la sección 3.1.3. Las ecuaciones para obtener los elementos de las matrices L y U son semejantes en los tres métodos antes vistos, esto es porque la base de solución es la misma, encontrar dos matrices triangulaes inferior y superior las cuales al multiplicarlas se obtenga la matriz original.

5. Compilación y ejecución de los programas

5.1. Factorización LU - Crout

Los programas contenidos en las carpetas Problema_1 y Problema_2 son compilados con el siguiente comando:

```
gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -std=c11
```

Para ejecutar los programas se usa el siguiente comando:

```
./main.out matrix vector
```

donde matrix es el nombre del archivo que contiene a los datos de la matriz y vector es el archivo que contiene los datos del vector columna. En el caso de la carpeta Problema_1 estos archivos son (test_matrix.txt,test_result.txt) y (M_LARGE.txt,V_LARGE.txt). Para la carpeta Problema_2 los archivos son (M_SMALL.txt,V_SMALL.txt)

5.2. Factorización Chelosky

Los programas contenidos en las carpetas Problema_3_test y Problema_3_1 son compilados con el siguiente comando:

```
gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -std=c11 -lm
```

5.3. Matriz de prueba

Para ejecutar los programas de la carpeta Problema_3_test se usa el siguiente comando:

```
./main.out matrix l_matrix lt_matrix
```

donde matrix es el nombre del archivo que contiene a los datos de la matriz , l_matrix y lt_matrix son los archivos donde se guardaran las matrices L y L^T. En este caso los archivos reciben los siguientes nombres test_matrix.txt, L_test.txt y LT_test.txt.

5.4. Matrices dadas

Para ejecutar los programas de la carpeta Problema_3_1 se usa el siguiente comando:

```
./main.out n l_matrix lt_matrix
```

donde n es un número entero el cual dara el tamaño de la matriz, l_matrix y lt_matrix son los archivos donde se guardaran las matrices L y L^T.
Si se quiere una matrix de 4×4, la cual sus matrices L y L^T se guarden en los archivos L_4.txt y LT_4.txt el comando deberia ser el siguiente:

```
./main.out 4 L_4.txt LT_4.txt
```

Se creo un script en bash el cual ejecuta los valores de n comentados en la sección 3.2.2.