Centro de Investigación en Matemáticas A. C.

Maestría en Ciencias con especialidad en Computación y Matemáticas Industriales

Métodos Numéricos (2019)

Tarea 03

Reporte

Marco Antonio Esquivel Basaldua (MCMESQ)

El presente reporte de tarea muestra y explica distintos algoritmos utilizados para la solución de sistemas de ecuaciones lineales así como una forma de encontrar el valor absoluto máximo en una matriz y su localización (fila y columna), y una forma de encontrar la inversa de una matriz cuadrada dada siempre y cuando esta sea invertible.

Para la solución de sistemas de ecuaciones lineales se consideran los casos de dar solución a una matriz diagonal de coeficientes, una matriz triangular superior, una matriz triangular inferior y de forma general se utiliza el método de eliminación gaussiana en dos versiones: operaciones sin pivoteo y operaciones con pivoteo. Algoritmos de factorización, previos a la solución de los sistemas, también son desarrollados: factorización A = LU por el algoritmo de Doolittle y la factorización $A = LDL^T$ por el algoritmo de Cholesky.

Los códigos utilizados para estas aplicaciones están desarrollados en el lenguaje C y creados y compilados en el entorno de desarrollo Visual Studio Code.

1. Solución de sistemas de ecuaciones

Al tener un sistema de ecuaciones lineales el objetivo principal es obtener los valores de las variables x_i que dan solución a cada una de las ecuaciones en el sistema. Para el presente trabajo se van a considerar únicamente sistemas de ecuaciones con el mismo número de incógnitas que de ecuaciones. Dicho sistema de ecuaciones se puede representar en forma de multiplicación de matrices de la forma Ax = b, con la matriz A como la matriz de coeficientes, el vector x como el vector de incógnitas y el vector b como el vector de soluciones a cada una de las ecuaciones:

Sistema de ecuaciones lineales:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n$$

Su representación en la forma de multiplicación de matrices se da de la siguiente forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Dadas las diferentes configuraciones para la matriz A de coeficientes se puede desarrollar algoritmos para encontrar las soluciones a las incógnitas. Dichos algoritmos se explican a continuación.

a. Matriz Diagonal

Dado un sistema de ecuaciones lineales cuya matriz de coeficientes es una matriz diagonal que cuenta con ceros en sus entradas salvo por las entradas en la diagonal los cuales deben ser forzosamente distintos de cero, la solución de las incógnitas se realiza de la siguiente manera:

Representación del sistema en la forma Ax = b:

$$\begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Al realizar la multiplicación de matrices los resultados son:

$$a_{11}x_1 = b_1$$

$$a_{22}x_2 = b_2$$

$$a_{33}x_3 = b_3$$

$$a_{nn}x_n = b_n$$

Algoritmo de resolución:

Para x_1:

$$x_1 = \frac{b_1}{a_{11}}$$

Para x_2:

$$x_2 = \frac{b_2}{a_{22}}$$

De forma general para x_i donde i = 1, 2, …, n

$$x_i = \frac{b_i}{a_{ii}}$$

Para el cálculo del determinante es suficiente multiplicar las entradas a_{ii} de la matriz A:

$$\det(A) = a_{11} * a_{22} * a_{33} * \dots * a_{nn}$$

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_DIAG.txt
- Leer y guardar la matriz b desde el archivo V_DIAG.txt
- Se evalúa si el algoritmo puede ser implementado
  - A debe ser cuadrada y diagonal
  - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
  - Se calcula cada una las x_i de acuerdo al algoritmo antes citado
  - Se calcula el determinante de A
  - Se imprimen en pantalla las solucione de x y el determinante de A.

b. Matriz triangular superior

Dado un sistema de ecuaciones lineales cuya matriz de coeficientes es una matriz triangular superior cuyas entradas en la diagonal las cuales deben ser forzosamente distintas de cero, la solución de las incógnitas se realiza de la siguiente manera:

Representación del sistema en la forma Ax = b:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Al realizar la multiplicación de matrices los resultados son:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2$$

$$a_{33}x_3 + \cdots + a_{3n}x_n = b_3$$

$$a_{nn}x_n = b_n$$

Algoritmo de resolución:

Primero se calcula la última de las incógnitas:

$$x_n = \frac{b_n}{a_{nn}}$$

Se calculan los valores de x de manera regresiva, es decir desde x_n hasta x_1

$$x_{n-1} = \frac{b_{n-1} - a_{(n-1)n} * x_n}{a_{(n-1)(n-1)}}$$

$$x_{n-2} = \frac{b_{n-2} - a_{(n-2)(n-1)} * x_{n-1} - a_{(n-1)n} * x_n}{a_{(n-1)(n-1)}}$$

De manera general para $i = n - 1, n - 2, \dots 1$

$$x_i = \frac{b_i - \sum_{j=i+n}^{n} a_{ij} * x_j}{a_{ii}}$$

Para el cálculo del determinante es suficiente multiplicar las entradas a_{ii} de la matriz A:

$$\det(A) = a_{11} * a_{22} * a_{33} * \dots * a_{nn}$$

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_TSUP.txt
- Leer y guardar la matriz b desde el archivo V_TSUP.txt
- Se evalúa si el algoritmo puede ser implementado
  - A debe ser cuadrada y triangular superior
  - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
  - Se calcula cada una las x_i de acuerdo al algoritmo antes citado
    - Se calcula x_n
    - Para i = n-1, n-2, … 1
      - Se calcula en la variable sum el valor de la sumatoria
      - Se calcula x_i
      - Reinicializar sum a cero
  - Se calcula el determinante de A
  - Se imprimen en pantalla las solucione de x y el determinante de A.

c. Matriz triangular Inferior

Dado un sistema de ecuaciones lineales cuya matriz de coeficientes es una matriz triangular inferior cuyas entradas en la diagonal las cuales deben ser forzosamente distintas de cero, la solución de las incógnitas se realiza de la siguiente manera:

Representación del sistema en la forma Ax = b:

$$\begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Al realizar la multiplicación de matrices los resultados son:

$$a_{11}x_1 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n$$

Algoritmo de resolución:

Primero se calcula la primera de las incógnitas:

$$x_1 = \frac{b_1}{a_{11}}$$

Se calculan los valores de x de manera progresiva, es decir desde x_1 hasta x_n

$$x_2 = \frac{b_2 - a_{21} * x_1}{a_{22}}$$

$$x_3 = \frac{b_3 - a_{31} * x_1 - a_{32} * x_2}{a_{33}}$$

De manera general para $i = 2, 3, 4 \ldots, n$

$$x_i = \frac{b_i - \sum_{j=1}^{n} a_{ij} * x_j}{a_{ii}}$$

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_TINF.txt
- Leer y guardar la matriz b desde el archivo V_TINF.txt
- Se evalúa si el algoritmo puede ser implementado
    - A debe ser cuadrada y triangular inferior
    - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
    - Se calcula cada una las x_i de acuerdo al algoritmo antes citado
        - Se calcula x_1
        - Para i = 2, 3, …, n
            - Se calcula en la variable sum el valor de la sumatoria
            - Se calcula x_i
            - Reinicializar sum a cero
    - Se calcula el determinante de A
    - Se imprimen en pantalla las solucione de x y el determinante de A.

d. Solución por eliminación gaussiana

Para este caso la representación y desarrollo del sistema de ecuaciones es el siguiente:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n$$

Como se puede apreciar no existe una forma directa de calcular alguna de las x_i dados únicamente la matriz A y el vector b por lo que se debe buscar una forma de dar solución a este tipo de sistemas de ecuaciones.

Algoritmo de resolución:

Una forma de solucionar el sistema es, en una primera instancia, transformando la matriz A en su equivalente triangular superior realizando operaciones por fila incluyendo al vector b por lo que las operaciones deben ser realizadas en la matriz aumentada A|b de la siguiente forma:

Para la primera fila de la matriz aumentada: esta no cambia.

Para la segunda fila sus elementos se calculan como sigue (el asterisco significa que el elemento ha sido transformado y colocado nuevamente en esa posición):

$$a_{21}^* = 0$$

$$a_{22}^* = a_{22} - a_{21}\frac{a_{12}}{a_{11}}$$

$$a_{23}^* = a_{23} - a_{21}\frac{a_{13}}{a_{11}}$$

$$a_{2i}^* = a_{2i} - a_{21}\frac{a_{1i}}{a_{11}}$$

Para la tercera fila:

$$a_{31}^* = 0$$

$$a_{32}^* = a_{32} - a_{31}\frac{a_{12}}{a_{11}}$$

$$a_{33}^* = a_{33} - a_{31}\frac{a_{13}}{a_{11}}$$

$$a_{3i}^* = a_{3i} - a_{31}\frac{a_{1i}}{a_{11}}$$

El proceso se repite hasta llegar a la última fila.

Los pasos anteriores se repiten recorriendo las columnas y filas en orden pero sin modificar las filas que ya tienen un cero en la posición deseada hasta obtener un sistema de la siguiente forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{*} & a_{23}^{*} & \cdots & a_{2n}^{*} \\ 0 & 0 & a_{33}^{**} & \cdots & a_{3n}^{**} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{*} \\ b_3^{*} \\ \vdots \\ b_n^{(n)} \end{bmatrix}$$

Una vez obtenida la matriz anterior se resuelve el sistema de ecuaciones como si se tratara de una sistema con una matriz triangular superior tal y como se explicó en la sección b de este apartado.

El determinante de A se obtiene al multiplicar los elementos en la diagonal de la matriz triangular obtenida.

Nota: ya que no se realizan pivoteos, este algoritmo es propenso a fallar si se llega a un cero en la diagonal antes de llegar a la última de las filas. Este problema se soluciona en el algoritmo de la sección e de este apartado.

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_SMALL.txt
- Leer y guardar la matriz b desde el archivo V_ SMALL.txt
- Se evalúa si el algoritmo puede ser implementado
    - A debe ser cuadrada
    - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
    - Se transforma la matriz aumentada a una matriz triangular superior
    - Se calcula el determinante de A
    - Si el determinante es posible de calcular y diferente de cero.
        - Se calculan los valores de x utilizando el algoritmo para matrices triangulares superiores
        - Se imprimen en pantalla las solucione de x y el determinante de A.

e. Solución por eliminación gaussiana y pivoteo

Esta sección muestra un algoritmo extendido para el proceso de eliminación gaussiana que asegura su correcta implementación y otorga más precisión a los resultados. La mejora al algoritmo anterior radica en el pivoteo de filas y columnas que va colocando el elemento en A con mayor valor en la diagonal según se vaya avanzando en la ejecución del programa.

Algoritmo de resolución:

Se localiza el elemento en A con mayor valor absoluto (este proceso se muestra en el apartado 2) y se mueve a la posición a_{11} intercambiando filas y columnas (Nota: si se intercambian columnas se estarán cambiando el orden de las x_i al ser calculadas, esto debe ser tomado en cuenta al momento de mostrar las soluciones de x).

Una vez cambiadas filas y columnas se encuentran los ceros en los elementos a_{21}, a_{31},… a_{n1} tal como se planteó en la sección d.

El proceso se repite encontrando el máximo valor absoluto de la matriz A exceptuando la búsqueda en las filas cuyos ceros ya han sido calculados hasta obtener la matriz triangular superior equivalente a A.

Se calculan los valores de x según el algoritmo para matrices triangulares superiores.

El determinante de A se obtiene al multiplicar los elementos en la diagonal de la matriz triangular obtenida.

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_SMALL.txt
- Leer y guardar la matriz b desde el archivo V_ SMALL.txt
- Se evalúa si el algoritmo puede ser implementado
  - A debe ser cuadrada
  - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
  - Se transforma la matriz aumentada a una matriz triangular superior usando pivoteo
  - Se calcula el determinante de A
  - Si el determinante es posible de calcular y diferente de cero.
    - Se calculan los valores de x utilizando el algoritmo para matrices triangulares superiores
    - Se imprimen en pantalla las solucione de x y el determinante de A.

f. Solución de sistema de ecuaciones usando factorización LU por el algoritmo de Dolittle

Dada una matriz cuadrada, A, esta se puede factorizar como la multiplicación de una matriz triangular inferior, L, y una matriz triangular superior, U, por lo que un sistema de ecuaciones puede escribirse de la forma:

$$Ax = b$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

$$LUx = b$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ L_{21} & 1 & 0 & \dots & 0 \\ L_{31} & L_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & L_{n3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} & \cdots & U_{1n} \\ 0 & U_{22} & U_{23} & \cdots & U_{2n} \\ 0 & 0 & U_{33} & \cdots & U_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & U_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Nótese que los elementos en la diagonal de L son todos 1.

Algoritmo de resolución:

Al tener la matriz A factorizada de esa forma se puede solucionar el siguiente sistema con el algoritmo para matrices triangulares inferiores:

$$Ly = b$$

Donde:

$$y = Ux$$

Una vez solucionada y se resuelve el sistema siguiente encontrando el valor de las x usando el algoritmo para matrices triangulares superiores:

$$Ux = y$$

El determinante de A se obtiene al multiplicar los elementos en la diagonal de la matriz U.

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_SMALL.txt
- Leer y guardar la matriz b desde el archivo V_ SMALL.txt
- Se evalúa si el algoritmo puede ser implementado
  - A debe ser cuadrada
  - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
  - Se factoriza A en las matrices L y U
  - Se calcula el determinante de A
  - Si el determinante es posible de calcular y diferente de cero.
    - Se calculan los valores de y a partir del sistema Ly = b
    - Se calculan los valores de x a partir del sistema Ux = y
    - Se imprimen en pantalla las solucione de x y el determinante de A.

g. Solución de sistema de ecuaciones usando factorización LDL^T por el algoritmo de Cholesky

Dada una matriz cuadrada, A, esta se puede factorizar como la multiplicación de una matriz triangular inferior, L, una matriz diagonal, D, y la transpuesta de L por lo que un sistema de ecuaciones puede escribirse de la forma:

$$Ax = b$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

$$LDL^T x = b$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ L_{21} & 1 & 0 & \dots & 0 \\ L_{31} & L_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & L_{n3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} D_{11} & 0 & 0 & \dots & 0 \\ 0 & D_{22} & 0 & \dots & 0 \\ 0 & 0 & D_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D_{nn} \end{bmatrix} \begin{bmatrix} 1 & L_{21} & L_{31} & \dots & L_{n1} \\ 0 & 1 & L_{32} & \dots & L_{n2} \\ 0 & 0 & 1 & \cdots & L_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Nótese que los elementos en la diagonal de L y L^T son todos 1.

Algoritmo de resolución:

Al tener la matriz A factorizada de esa forma se puede solucionar el siguiente sistema con el algoritmo para matrices triangulares inferiores:

$$Lz = b$$

Donde:

$$z = Dy$$

Una vez solucionada y se resuelve el sistema siguiente encontrando el valor de las x usando el algoritmo para matrices diagonales:

$$Dy = z$$

Donde:

$$y = L^T x$$

Una vez solucionada y se encuentran los valores para x aplicando el algoritmo para matrices triangulares superiores:

$$L^T x = y$$

El determinante de A se obtiene al multiplicar los elementos en la diagonal de la matriz D.

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_SMALL.txt
- Leer y guardar la matriz b desde el archivo V_ SMALL.txt
- Se evalúa si el algoritmo puede ser implementado
  - A debe ser cuadrada
  - b debe tener tantos valores como filas en la matriz A
- Si el algoritmo puede ser implementado:
  - Se factoriza A en las matrices L y U

    - Se calcula el determinante de A
    - Si el determinante es posible de calcular y diferente de cero.
        - Se calculan los valores de z a partir del sistema Lz = b
        - Se calculan los valores de y a partir del sistema Dy = z
        - Se calculan los valores de x a partir del sistema $L^T x = y$
        - Se imprimen en pantalla las solucione de x y el determinante de A.

2. Máximo valor absoluto en una función

Dada una matriz A se localiza el elemento con mayor valor absoluto y se registra su posición.

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_MEDUIM.txt
- Se declara una variable *max* y se inicializa a cero.
- Se declaran dos variables *i_ind* y *j_ins* donde se registrarà la posición del valor máximo
- Se rocorren los elementos en A
    - Si se encuentra un elemento cuyo valor absoluto sea mayor a max:
        - max toma ese valor y se registra su posición en filas y columnas en *i_ind* y *j_ins*

3. Inversa de una matriz

A partir de una matriz cuadrada dada se puede calcular su inversa primero factorizando A = LU mediante al algoritmo de Doolittle.

Implementación en código C, pseudocódigo:

- Leer y guardar la matriz A desde el archivo M_MEDUIM.txt
- Se declara una matriz A_inv y se inicializa como la matriz identidad.
- Se realiza la factorización LU para A
- Se resuelve la ecuación Ux = A_inv(i) donde A_inv(i) es cada una de las columnas en A_inv
    - El vector x remplaza cada una de las columnas en A_inv que se van calculando
- El proceso se repite solucionanado para Lx = A_inv(i)
- Para verificar la solución se multiplican A * A_inv y el resultado debe ser la matriz identidad.