Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Asignatura: Cálculo Científico (6105)

Estudiante: Naranjo Sthory Alexanyer Antonio

Cédula de identidad: V – 26.498.600

Tarea 1: Factorización LU y Resolución de SEL

A continuación, se presentan las respuestas de cada una de las preguntas indicadas para la actual asignación, donde se ha separado cada una según sus respectivos incisos, además de adjuntar la correspondiente justificación a la respuesta de cada una de las preguntas.

Se destaca también la carpeta cuyo nombre es ***Codes***, que contiene el código fuente de la resolución de aquellos ejercicios que lo requieren. De igual manera, en el presente informe se indican aquellas preguntas que se solventaron a partir de una implementación en Matlab/Octave y también se adjuntan imágenes de aquellos fragmentos de código relevantes para la justificación de la respuesta.

Respuestas a):

La mayoría de preguntas relacionadas a este inciso consisten de **Verdadero** o **Falso**. Para ello, se ha indicado en negrita y color rojo la opción indicada y justamente debajo de cada pregunta, se encuentra la justificación para dicha respuesta.

2.2. **Verdadero** o Falso: Si una matriz tiene un determinante muy pequeño, entonces la matriz es casi singular.

Justificación: Este fenómeno se le denomina **Colinealidad**, Donde una variable X_1 sea combinación lineal de otra X_2, significa que ambas están relacionadas por la expresión $X_1 = \beta_1 + \beta_2 X_2$, siendo β_1 y β_2 son constantes, por lo tanto, el coeficiente entre ambas variables será 1.

Del mismo modo, que una variable X_1, sea combinación lineal de otras $X_2, \dots, X_i$ con $i > 2$, significa que dichas variables están relacionadas por la expresión $X_1 = \beta_1 > + \beta_2 X_2 + \dots + \beta_i X_i$, siendo $\beta_i, \dots, \beta_i$ constantes y, por tanto, el coeficiente de correlación múltiple $R_{x_1|x_2|\dots x_i}$ también será 1.

En la práctica, esta colinealidad exacta raras veces ocurre, pero sí surge con cierta frecuencia la llamada *casi colinealidad*, o por extensión, simplemente colinealidad en que alguna variable es "casi" combinación lineal de otra u otras, o, dicho de otro modo, algunos coeficientes de correlación simple o múltiple entre las variables independientes están cercanos a 1, aunque no llegan a dicho valor.

En este caso la matriz $X'X$ es *casi singular*, es decir su determinante no es cero, pero es muy pequeño. Como para invertir una matriz hay que dividir por su determinante, en esta situación surgen problemas de precisión en la estimación de los coeficientes, ya que los algoritmos de inversión de matrices pierden precisión al tener que dividir por un número muy pequeño, siendo además inestables.

2.3. **Verdadero** o Falso: Si una matriz triangular tiene una entrada cero en su diagonal principal, entonces la matriz es necesariamente singular.

Justificación: El determinante se puede calcular con facilidad utilizando la matriz triangulada. El determinante de los elementos triangulados es el producto de los elementos diagonales. Como los elementos diagonales son cero, la matriz es singular.

Consideremos la siguiente matriz triangular superior 3x3 **A**, expresada en su forma general y con un valor nulo (cero) en su diagonal principal calculemos su respectivo determinante, recordando que una matriz singular es aquellas que, entre sus condiciones, cumple que $det(A) = 0$.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}$$

Tenemos que,

$$det(A) = a_{11}(0) - a_{12}(0) + a_{13}(0) = 0$$

Dado a que $det(A) = 0$, entonces, la matriz **A** indicada es singular.

2.4. Verdadero o **Falso**: Si una matriz tiene una entrada nula en su diagonal principal, entonces la matriz es necesariamente singular.

Justificación: Para probar que es Falso esta suposición, basta con encontrar un contraejemplo que apoye lo indicado. Para ello, consideremos la siguiente matriz 3 x 3 **A**,

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 0 & 4 \\ 5 & 1 & 2 \end{pmatrix}$$

Si calculamos el respectivo determinando de la matriz **A**, obtendremos el siguiente resultado,

$$\det(A) = 1(0 - 4) - 2(6 - 20) + 3(3 - 0)$$

$$= -4 - 2(-14) + 9$$

$$= -4 + 28 + 9$$

$$\det(A) = 33$$

Dado a que $\det(A) \neq 0$, entonces, se concluye que si tenemos una matriz cualquiera **A,** la cual contiene una entrada nula en su diagonal principal, no necesariamente es singular.

2.6. **Verdadero** o Falso: El producto de dos matrices triangulares superiores es triangular superior.

Justificación: Consideremos dos matrices n x n **A** y **B** triangulares superiores y $\boldsymbol{C} = \boldsymbol{AB}$. Entonces, considerando la entrada C_{ij} de **C** con $i > j$, tenemos que,

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$$= \sum_{k=1}^{i-1} a_{ik} b_{kj} + \sum_{k=i}^{n} a_{ki} b_{kj}$$

$$= 0 + 0 = 0,$$

Donde,

$$k \leq \mathrm{i} - 1 \Longrightarrow a_{ik} = 0$$

$$k \geq i \Longrightarrow b_{kj} = 0$$

Para esta particular combinación de (i, j). Podemos concluir que $C_{ij} = 0$, si $i > j$, por lo tanto, **C** es triangular superior.

Podemos generalizar la afirmación anterior de la siguiente manera,

Sean $U_1, U_2, \dots, U_k$ matrices triangulares superiores n x n. Entonces,

$$V = \prod_{i=1}^{k} U_i$$

Es una matriz triangular superior n x n.

2.9. **Verdadero** o Falso: Si las filas de una matriz n × n son linealmente dependientes, entonces las columnas de la matriz también son linealmente dependientes.

Justificación: Si las filas de una matriz n x n **A** son linealmente independientes, entonces el resultado de hacer la reducción de fila para **A** es la matriz identidad, entonces la única solución de $\boldsymbol{Av} = \mathbf{0}$ es $\boldsymbol{v} = \mathbf{0}$.

Si las columnas de **A** son linealmente dependientes, se dice que,

$$a_1c_1 + a_2c_2 + \dots + a_nc_n = 0,$$

Donde c_i son las columnas y a_i no son todos ceros, entonces $Av = 0$, donde,

$$V = (a_1, a_2, \dots, a_n) \neq 0.$$

Entonces, si las columnas son linealmente dependientes, entonces también lo serán las filas.

Si aplicamos este argumento a la matriz transpuesta de **A,** donde definamos que las filas son linealmente dependientes, queda demostrado que las columnas también lo serán.

Un argumento más intuitivo (por qué para una matriz cuadrada (n x n), si sus filas son linealmente dependientes implica que también lo son sus columnas). Dado que las filas son linealmente dependientes, la realización de la eliminación de Gauss debe producir una fila de todos los ceros (posiblemente más de uno). Por lo tanto, las columnas de la matriz reducida fila de la forma del escalón son linear dependientes. Eso es así porque todos tienen cero en la misma entrada, por lo tanto, solo tienen n - 1 variables "libres", entradas no nulas, por lo tanto, son n - 1 dimensionales, y n vectores en solo n - 1 espacio dimensional- no pueden ser linealmente independientes.

Y dado que las columnas de la matriz reducida son linealmente dependientes, por lo que deben ser también las columnas de la matriz original, que es nuestra afirmación.

2.12. **Verdadero** o Falso: Siempre que los intercambios de filas estén permitidos, la factorización LU siempre existe incluso para una matriz singular **A**.

Justificación: Si A es una matriz singular de rango k, entonces admite una factorización LU si los primeros k menores principales son distintos de cero, aunque lo contrario no es cierto.

2.13. **Verdadero** o Falso: Si un sistema lineal está bien condicionado, entonces el pivoteo es innecesario en eliminación gaussiana.

Justificación: Realizar el pivoteo puede introducir errores en los valores de la eliminación gaussiana, por lo que es innecesario para un sistema lineal bien condicionado, solamente será necesario aplicar este método siempre que tengamos con nosotros un sistema de ecuaciones lineales $Ax = b$ mal condicionado.

Recordemos, a nivel de Sistema de Ecuaciones Lineales, diremos que un sistema es "sensible" o también llamada "mal condicionados", si ante una pequeña variación de los valores de los coeficientes de la matriz de coeficientes **A**, del vector de términos independiente **b**, o de ambos, se produce una variación importante en los valores de la solución del mismo.

Este tipo de sistemas, se caracterizan por poseer una matriz **A**, con un numero de condición:

$$Cond(A) > 1$$

2.14. Verdadero o **Falso**: Si una matriz es singular entonces no puede tener una factorización LU.

Justificación: Sí, una matriz singular puede tener una descomposición LU. Una condición suficiente para una matriz de rango **k**, es que los primeros **k** menores principales de la matriz sean distintos de cero.

Vamos a crear un ejemplo de 3x3. Supongamos que queremos,

$$\begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33} \end{bmatrix}$$

para ser una matriz no invertible. El determinante de la primera matriz es $a_{11}a_{22}a_{33}$ y el determinante de la segunda matriz es $b_{11}b_{22}b_{33}$, así que, si el producto tiene determinante cero, una de esas entradas tiene que ser cero. Escojamos $b_{22} = 0$ y rellenemos todo lo demás con 1's:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{bmatrix}$$

Se trata de una matriz singular, de rango 2, pero la hemos obtenido multiplicando una matriz triangular inferior y otra superior, por lo que debe tener una descomposición LU.

2.17. **Verdadero** o Falso: La eliminación gaussiana sin pivotar sólo falla cuando la matriz está mal condicionada o es singular.

Justificación: Para una matriz cuadrada n x n, la eliminación gaussiana fallará si el determinante es cero. Para una matriz arbitraria, fallará si alguna fila es una combinación lineal de las filas restantes, aunque se puede cambiar el problema eliminando dichas filas y hacer la reducción de filas en la matriz restante. La eliminación gaussiana con pivotaje completo (intercambio de filas o columnas para tener el mayor valor absoluto posible de cada elemento "pivote") fallará sólo en esas circunstancias. Sin embargo, la eliminación gaussiana ingenua sin pivote **a veces fallará** (dividir por cero) en una matriz perfectamente bien condicionada.

2.18. **Verdadero** o Falso: Una vez que la factorización LU de una matriz **A** se ha calculado para resolver un sistema lineal **Ax = b**, los siguientes sistemas lineales con la misma matriz, pero con diferentes vectores del lado derecho pueden resolverse sin refactorizar la matriz.

Justificación: En ocasiones, la factorización LU se aplica como una rutina simple, pero esta puede dividirse para realizar dos tareas o rutinas diferentes:

- Una de ellas para calcular una correspondiente factorización a una matriz **A** dada.
- La segunda para resolver un sistema de ecuaciones con el sistema triangular resultante.

La factorización o, mejor dicho, refactorización, no debería de ser necesaria ya que se necesitan soluciones adicionales con la misma matriz, pero con un vector **b** derecho diferente. Es decir, basta con calcular la factorización LU de una matriz **A** y dicho resultado, es capaz de trabajar sin inconveniente alguno para cualquier vector **b** derecho que se establezca.

Este es una de las ventajas principales que presenta realizar esta factorización dado a que podremos resolver cualquier sistema de ecuaciones que tenga a la matriz de coeficientes **A** y cualesquiera valores para el vector derecho **b**.

2.25. **Verdadero** o Falso: Al resolver un sistema de ecuaciones lineales, la eliminación gaussiana con pivoteo parcial suele producir un residual incluso si la matriz está mal condicionada.

Justificación: Podemos probar que esto es cierto a partir de la resolución de un ejemplo que apoye la opción seleccionada.

Consideremos la resolución del siguiente sistema de ecuaciones con y sin pivoteo.

$$\begin{cases} 0.129x + 0.81y + 0.9z = 0.6867 \\ x + y + z = 0.8338 \\ 1.331x + 1.21y + 1.1z = 1.0000 \end{cases}$$

Donde,

$$x = 0.2249; y = 0.2814; z = 0.3219$$

Calculemos la solución sin hacer uso de pivoteo, basándonos únicamente de cuatro (4) decimales aritméticos.

$$\begin{pmatrix} 0.1290 & 0.8100 & 0.9000 & | & 0.6867 \\ 1.0000 & 1.0000 & 1.0000 & | & 0.8338 \\ 1.3310 & 1.2100 & 1.1000 & | & 1.0000 \end{pmatrix}$$

$\rightarrow m_{21} = 1.312; m_{31} = 1.826$

$$\begin{pmatrix} 0.1290 & 0.8100 & 0.9000 & | & 0.6867 \\ 0.0000 & -0.1110 & -0.2350 & | & -0.1084 \\ 0.0000 & -0.2690 & -0.5430 & | & -0.2540 \end{pmatrix}$$

$\rightarrow m_{32} = 2.423$

$$\begin{pmatrix} 0.1290 & 0.8100 & 0.9000 & | & 0.6867 \\ 0.0000 & -0.1110 & -0.2350 & | & -0.1084 \\ 0.0000 & 0.0000 & -0.02640 & | & 0.008160 \end{pmatrix}$$

Realizando la sustitución hacía atrás, obtenemos que,

$$x = 0.2251; y = 0.2190; z = 0.3295$$

Observando que el uso del pivoteo trae consigo un erro al momento de realizar la Eliminación Gaussiana y concluyendo en que la oración indicada es Verdadera.

2.30.

a) Indique una propiedad que defina una matriz singular A.

Justificación: Si **A** es una matriz cuadrada singular, entonces, $|A| = 0$.

b) Suponga que el sistema lineal **Ax = b** tiene dos soluciones distintas **x** e **y**. Utilice la propiedad que dio en la parte a para demostrar que A debe ser singular.

Justificación: Como, $Ax = B$ tiene dos soluciones distintas, entonces, $x = \vec{A}b$, y $\vec{A}$ existen únicamente cuando $|A| \neq 0$.

Por ejemplo, **A** debe ser una matriz singular para,

$$\begin{cases} \mathbf{2x + 3y = 5} \\ \mathbf{2x + y = 3} \end{cases}$$

Donde,

$$\mathbf{A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} \end{pmatrix} \Rightarrow |\mathbf{A}| = \mathbf{2} - \mathbf{6} = -\mathbf{4} \Rightarrow \mathbf{A}^{-1} = -\frac{\mathbf{1}}{\mathbf{4}} \begin{pmatrix} \mathbf{1} & -\mathbf{3} \\ -\mathbf{2} & \mathbf{2} \end{pmatrix}$$

Entonces,

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = -\frac{\mathbf{1}}{\mathbf{4}} \begin{pmatrix} \mathbf{1} & -\mathbf{3} \\ -\mathbf{2} & \mathbf{2} \end{pmatrix} \begin{pmatrix} \mathbf{5} \\ \mathbf{3} \end{pmatrix} = -\frac{\mathbf{1}}{\mathbf{4}} \begin{pmatrix} -\mathbf{4} \\ -\mathbf{4} \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \end{pmatrix}$$

Por lo tanto, tendrá (x, y) = (1, 1) como dos soluciones distintas.

Allí, el sistema tiene soluciones distintas únicamente cuando $|A| \neq 0$, y para el caso contrario, el sistema tendrá infinitas soluciones o no tendrá ninguna.

2.41. Dada una matriz n x n no singular **A** y una segunda matriz n x n **B**, ¿Cuál es la mejor manera de calcular la matriz n x n $\boldsymbol{A^{-1}B}$?

Justificación: En primer lugar, tenemos que saber exactamente cuándo estamos hablando sobre una matriz singular y cuando no.

Recordemos que una matriz singular es aquella matriz cuadrada n x n cuyo determinante es igual a cero (0), en caso contrario, se dice que es la matriz es no singular.

Por lo tanto, dado una matriz cuadrada n x n **A** no singular.

Sabemos que para toda matriz cuadrada, esta posee su respectiva inversa, digamos A^{-1}, y,

$$\boldsymbol{A^{-1}} = \frac{\boldsymbol{adj}(\boldsymbol{A})}{\mathbf{det}(\boldsymbol{A})} = \frac{\boldsymbol{adj}(\boldsymbol{A})}{|\mathbf{A}|}$$

Primero, debemos hallar A-1 y luego la multiplicación respectiva con la segunda matriz n x n **B**.

Para A^{-1}, necesitaremos hallar $adj(A)$, así que tomemos una matriz no singular como ejemplo para llevar a cabo el procedimiento.

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & -\mathbf{2} \\ -\mathbf{1} & \mathbf{3} & \mathbf{5} \\ \mathbf{2} & \mathbf{7} & -\mathbf{4} \end{pmatrix}$$

Y procedamos a calcular su respectivo determinante,

$$\mathbf{det}(\boldsymbol{A}) = \mathbf{1}(-\mathbf{12} - \mathbf{35}) - \mathbf{2}(\mathbf{4} - \mathbf{10}) - \mathbf{2}(-\mathbf{7} - \mathbf{6})$$

$$= -\mathbf{47} + \mathbf{12} + \mathbf{26}$$

$$\mathbf{det}(\boldsymbol{A}) = -\mathbf{9}$$

Donde concluimos que existe A^{-1}, y procedemos a encontrar $adj(A)$. Para ello, necesitaremos encontrar los cofactores de los elementos.

$$\mathbf{A_{11}} = (-\mathbf{1})^{\mathbf{1+1}} \begin{vmatrix} \mathbf{3} & \mathbf{5} \\ \mathbf{7} & -\mathbf{4} \end{vmatrix} = (-\mathbf{12} - \mathbf{35}) = -\mathbf{47}$$

$$\mathbf{A_{12}} = (-\mathbf{1})^{\mathbf{1+2}} \begin{vmatrix} -\mathbf{1} & \mathbf{5} \\ \mathbf{2} & -\mathbf{4} \end{vmatrix} = -(\mathbf{4} - \mathbf{10}) = \mathbf{6}$$

$$\mathbf{A_{13}} = (-\mathbf{1})^{\mathbf{1+3}} \begin{vmatrix} -\mathbf{1} & \mathbf{3} \\ \mathbf{2} & \mathbf{7} \end{vmatrix} = (-\mathbf{7} - \mathbf{6}) = -\mathbf{13}$$

$$\mathbf{A_{21}} = (-\mathbf{1})^{\mathbf{2+1}} \begin{vmatrix} \mathbf{2} & -\mathbf{2} \\ \mathbf{7} & -\mathbf{4} \end{vmatrix} = -(-\mathbf{8} + \mathbf{14}) = -\mathbf{6}$$

$$\mathbf{A_{22}} = (-\mathbf{1})^{\mathbf{2+2}} \begin{vmatrix} \mathbf{1} & -\mathbf{2} \\ \mathbf{2} & -\mathbf{4} \end{vmatrix} = (-\mathbf{4} + \mathbf{4}) = \mathbf{0}$$

$$\mathbf{A_{23}} = (-\mathbf{1})^{\mathbf{2+3}} \begin{vmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{2} & \mathbf{7} \end{vmatrix} = -(\mathbf{7} - \mathbf{4}) = -\mathbf{3}$$

$$\mathbf{A_{31}} = (-\mathbf{1})^{\mathbf{3+1}} \begin{vmatrix} \mathbf{2} & -\mathbf{2} \\ \mathbf{3} & \mathbf{5} \end{vmatrix} = (\mathbf{10} + \mathbf{6}) = \mathbf{16}$$

$$\mathbf{A_{32}} = (-\mathbf{1})^{\mathbf{3+2}} \begin{vmatrix} \mathbf{1} & -\mathbf{2} \\ -\mathbf{1} & \mathbf{5} \end{vmatrix} = -(\mathbf{5} - \mathbf{2}) = -\mathbf{3}$$

$$\mathbf{A_{33}} = (-\mathbf{1})^{\mathbf{3+3}} \begin{vmatrix} \mathbf{1} & \mathbf{2} \\ -\mathbf{1} & \mathbf{3} \end{vmatrix} = (\mathbf{3} + \mathbf{2}) = \mathbf{5}$$

Luego, procedemos a calcular $Adj(A)$,

$$\mathbf{Adj}(\mathbf{A}) = \begin{pmatrix} -\mathbf{47} & \mathbf{6} & -\mathbf{13} \\ -\mathbf{6} & \mathbf{0} & -\mathbf{3} \\ \mathbf{16} & -\mathbf{3} & \mathbf{5} \end{pmatrix}^{\mathbf{T}} = \begin{pmatrix} -\mathbf{47} & -\mathbf{6} & \mathbf{16} \\ \mathbf{6} & \mathbf{0} & -\mathbf{3} \\ -\mathbf{13} & -\mathbf{3} & \mathbf{5} \end{pmatrix}$$

Finalmente, calculamos A^{-1},

$$\mathbf{A^{-1}} = \frac{\mathbf{Adj(A)}}{\mathbf{det(A)}} = -\frac{\mathbf{1}}{\mathbf{9}}\begin{pmatrix} \mathbf{-47} & \mathbf{-6} & \mathbf{16} \\ \mathbf{6} & \mathbf{0} & \mathbf{-3} \\ \mathbf{-13} & \mathbf{-3} & \mathbf{5} \end{pmatrix}$$

$$\mathbf{A^{-1}} = \begin{pmatrix} \mathbf{47/9} & \mathbf{2/3} & \mathbf{-16/9} \\ \mathbf{-2/3} & \mathbf{0} & \mathbf{1/3} \\ \mathbf{13/9} & \mathbf{1/3} & \mathbf{-5/9} \end{pmatrix}$$

Ahora, tomemos una matriz cuadrada **B** y realicemos la correspondiente multiplicación con la matriz A^{-1} hallada anteriormente.

$$\mathbf{B} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{3} & \mathbf{2} & \mathbf{3} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} \end{pmatrix}$$

$$\mathbf{A^{-1}B} = \begin{pmatrix} \mathbf{47/9} & \mathbf{2/3} & \mathbf{-16/9} \\ \mathbf{-2/3} & \mathbf{0} & \mathbf{1/3} \\ \mathbf{13/9} & \mathbf{1/3} & \mathbf{-5/9} \end{pmatrix}\begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{3} & \mathbf{2} & \mathbf{3} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} \end{pmatrix}$$

$$\mathbf{A^{-1}B} = \begin{pmatrix} \mathbf{49/9} & \mathbf{10} & \mathbf{11/3} \\ \mathbf{-1/3} & \mathbf{-1} & \mathbf{0} \\ \mathbf{17/9} & \mathbf{3} & \mathbf{4/3} \end{pmatrix}$$

Finalmente, este es el mejor camino aplicado para calcular $\boldsymbol{A^{-1}B}$.

Respuestas b):

De igual manera que en la sección anterior, se presentan las correspondientes respuestas a las preguntas relacionadas a la parte b) junto con sus respectivas justificaciones, además de adjuntar imágenes o cualquier tipo de apoyo que sea base para lo explicado en cada respuesta.

3.8.

Respuesta: Para el desarrollo de la resolución de este ejercicio, nos basaremos del código fuente desarrollado con el lenguaje de programación Octave, el cual será explicado a detalle a continuación.

Cabe destacar que varias funciones implementadas a lo largo de la resolución de este ejercicio, también serán utilizadas como respuesta para futuras preguntas correspondientes a la asignación, como por ejemplo la pregunta c) 5.6 y c) 5.9.

Primero se inicia con el desarrollo de la función **LU,** la cual se encargará de recibir una matriz cuadrada **A** y retornará los siguientes resultados:

1. **L**: La matriz cuadrada L, representa el factor izquierdo de la descomposición de la matriz A como multiplicación de dos matrices, recordando que dicha matriz L es triangular inferior.
2. **U**: Una matriz cuadrada superior, que corresponde al factor derecho de la descomposición de la matriz A.
3. **P**: Una matriz de permutación de la matriz original A. Esto, con el objetivo de demostrar que $PA = LU$.

El objetivo de obtener la descomposición LU de una matriz cuadrada A dada, es poder resolver cualquier Sistema de Ecuaciones Lineales de la forma $Ax = b$, y poder trabajar con cualquier vector **b**, es decir, tendremos la posibilidad de resolver cualquier SEL que contenga como matriz de coeficientes a la matriz **A.**

A continuación, se adjunta una imagen correspondiente al fragmento de código fuente que se encarga del cálculo de la descomposición LU, tomando en cuenta en un principio la inicialización de las matrices L y P con la función integrada **eye**, que se encarga de devolver una matriz identidad n x n, con el valor que nosotros le enviemos como argumento de entrada, e inicializamos la matriz U con los mismos valores originales que posee la matriz A que enviemos como argumento de entrada a la función LU definida propiamente.

Como en actividades anteriores, se ha realizado una intradocumentación del código para así facilitar la comprensión de cada una de las funciones definidas en la actividad.

```
function [L, U, P] = LU(A)
%A is the coefficient matrix
%L and U are lu factorized value of A
%P is parmutation matrix
%below is the algorithm for Gaussian elimination
    [m, n] = size(A);
    L = eye(n);
    P = eye(n);
    U = A;
    %initialization of L U and P
    %Loop for LU matrix formation for A matrix
    for k = 1:m-1
        pivot = max(abs(U(k:m,k)));
        for j = k:m
            if(abs(U(j,k)) == pivot)
                ind = j;
                break;
            endif
        endfor
        %updated L U and P values
        U([k,ind],k:m) = U([ind,k],k:m);
        L([k,ind],1:k-1) = L([ind,k],1:k-1);
        P([k,ind],:) = P([ind,k],:);
        %Loop for L and U matrix formation
        for j = k+1:m
            L(j,k) = U(j,k)/U(k,k);
            U(j,k:m) = U(j,k:m)-L(j,k)*U(k,k:m);
        endfor
    endfor
endfunction
```

Teniendo con nosotros la correspondiente factorización de la matriz **A,** entonces, procedemos a calcular las soluciones de cada una de los sistemas de ecuaciones indicados en el enunciado del ejercicio. Y para ello, se han implementado dos (2) funciones que se encargan propiamente de ello: **fsub** y **bsub.**

- **fsub**: Recibe como argumentos de entrada una matriz superior cuadrada L (la obtenida en la factorización LU realizada anteriormente) y un vector **b.** Esta función devolverá una solución **y** del sistema de ecuaciones $Ly = b$, realizando el proceso de sustitución hacía adelante.
- **bsub**: Recibe como argumentos de entrada una matriz inferior cuadrada U (la obtenida en la factorización LU realizada en pasos anteriores) y un vector **y.** Esta función devolverá una solución **x** del sistema de ecuaciones $Ux = y$, realizando el proceso de sustitución hacía atrás.

Como se menciona, podemos resolver un sistema de ecuaciones de la forma $Ax = b$ si conocemos la descomposición LU de la matriz de coeficientes A, partiendo de los factores L y U obtenidos, se procede a resolver dos (2) nuevos sistemas de ecuaciones.

1. $Ly = b$
2. $Ux = y$

Donde, los valores obtenidos para el vector **x** en el segundo paso, serán los valores para el sistema de ecuaciones original.

Se adjuntan las imágenes correspondientes a las funciones **fsub** y **bsub** mencionadas anteriormente.

```
function y = fsub(L,b)
        %Code for forward subtitution
        y = zeros(length(b),1);
        for i = 1:length(b)
        s = 0;
        for j = 1:length(b)
            if i ~= j
                s = s+L(i,j)*y(j);
            endif
        endfor
        y(i) = (b(i)-s)/L(i,i);
        endfor
endfunction
```

```
function x = bsub(U,y)
        %Code for Backward subtitution
        x = zeros(length(y),1);
        for i = length(y):-1:1
        s = 0;
        for j = length(y):-1:1
            if i ~= j
                s = s+U(i,j)*x(j);
            endif
        endfor
        x(i) = (y(i)-s)/U(i,i);
        endfor
endfunction
```

Por último, para cumplir con cada inciso de la pregunta actual, se realizó el proceso de refinamiento iterativo para la solución **x** calculada para el sistema de ecuaciones a tratar. Para ello, se definió la función **iter_ref,** que recibirá como argumentos de entradas:

- **A**: La matriz de coeficientes A correspondiente al Sistema de Ecuaciones Lineales a tratar.
- **b**: El vector derecho del sistema $Ax = b$.

Como se observará a continuación en la imagen adjunta de la función anteriormente mencionada, **iter_ref,** esta se ha implementado para que realice únicamente cuatro (4) iteraciones del refinamiento, obteniendo como valor de retorno el vector **x** de solución luego de haber hecho este proceso.

```
function x = iter_ref(A,b)
  [~,n] = size(A);
  x = inv(A)*b+1e-3*randn(n,1);
  [L,U] = LU(A);
  for k = 1:4
    r = A*x-b;
    z = -U\(L\r);
    x = x + z;
  endfor
endfunction
```

Se adjunta una imagen completa del código desarrollado para la resolución del ejercicio, recordando nuevamente que algunas funciones definidas en esta pregunta, serán nuevamente utilizadas en la resolución de un par de ejercicios a resolver próximamente.

```
function [L, U, P] = LU(A)
%A is the coefficient matrix
%L and U are lu factorized value of A
%P is parmutation matrix
%below is the algorithm for Gaussian elimination
    [m, n] = size(A);
    L = eye(n);
    P = eye(n);
    U = A;
    %initialization of L U and P
    %Loop for LU matrix formation for A matrix
    for k = 1:m-1
        pivot = max(abs(U(k:m,k)));
        for j = k:m
            if(abs(U(j,k)) == pivot)
                ind = j;
                break;
            endif
        endfor
        %updated L U and P values
        U([k,ind],k:m) = U([ind,k],k:m);
        L([k,ind],1:k-1) = L([ind,k],1:k-1);
        P([k,ind],:) = P([ind,k],:);
        %Loop for L and U matrix formation
        for j = k+1:m
            L(j,k) = U(j,k)/U(k,k);
            U(j,k:m) = U(j,k:m)-L(j,k)*U(k,k:m);
        endfor
    endfor
endfunction

function y = fsub(L,b)
  %Code for forward subtitution
  y = zeros(length(b),1);
  for i = 1:length(b)
  s = 0;
  for j = 1:length(b)
    if i ~= j
      s = s+L(i,j)*y(j);
    endif
  endfor
  y(i) = (b(i)-s)/L(i,i);
  endfor
endfunction

function x = bsub(U,y)
  %Code for Backward subtitution
  x = zeros(length(y),1);
  for i = length(y):-1:1
  s = 0;
    for j = length(y):-1:1
      if i ~= j
        s = s+U(i,j)*x(j);
      endif
    endfor
    x(i) = (y(i)-s)/U(i,i);
    endfor
endfunction

function x = iter_ref(A,b)
  [~,n] = size(A);
  x = inv(A)*b+1e-3*randn(n,1);
  [L,U] = LU(A);
  for k = 1:4
    r = A*x-b;
    z = -U\(L\r);
    x = x + z;
  endfor
endfunction

function main
  % Exercise (i)
  A = [0 1 0 0; 0 0 1 0; 0 0 0 1; 2 3 4 5];
  b = [1; 1; 1; 1];

  % Exercise (ii)
  % Matrix singular
  %A = [2 3 4 5; 0 1 0 0; 0 0 0 0; 0 0 1 0];
  %b = [1; 1; 1; 1];

  % Exercise (iii)
  %A = hilb(5);
  %b = [1; 1; 1; 1; 1];

  % Exercise (iv)
  %A = [10**(-4) 1; 1 2];
  %b = [1; 1];

  disp("LU decomposition")
  [L, U, P] = LU(A);
  disp("L = ")
  disp(L);
  disp("U = ")
  disp(U);
  disp("P = ")
  disp(P);
  disp("PA = ")
  disp(P * A)
  disp("LU = ")
  disp(L * U)
  disp("-------------------")

  disp("Solve system of linear equations Ax = b with LU decomposition")
  %First solve Ly = b with forward substitution, then solve Ux = y with backward substitution
  y = fsub(L, b);
  x = bsub(U, y);
  disp("Solution:")
  disp("x = ")
  disp(x)
  disp("-------------------")

  disp("Iterative Refinement - Solution ")
  if det(A) ~= 0
    x1 = iter_ref(A,b,x)
  else
    disp('A is singular')
  endif

endfunction

main()
```

Cabe mencionar que se ha definido también la función **main** como punto de partida de la ejecución del programa, además, dentro de ella estarán comentados cada uno de los casos a resolver para este ejercicio.

Respuestas c):

Se presenta a continuación, las respectivas respuestas de cada uno de los incisos de la pregunta c) de la tarea asignada. De igual manera a las respuestas de los incisos anteriores, se adjunta imágenes y/o visualizaciones de los fragmentos de código como apoyo en cada una de las explicaciones dada para las respuestas.

5.5. Indique que la factorización LU de A puede utilizarse para calcular la matriz inversa A^{-1}. (Obsérvese que el j-ésimo vector columna de A^{-1} satisface el sistema lineal A_{y_j} = e_j, siendo e_j el vector cuyas componentes son todas nulas excepto la j-ésima componente que es 1).

Respuesta: La descomposición o factorización LU es buena para resolver sistemas de ecuaciones lineales de la forma $Ax = B$, siendo **A** la misma matriz de coeficientes y **b** diferentes matrices. Esta es una excelente ventaja para calcular la inversa de la matriz **A**, dado a que únicamente requeriremos de una sola descomposición. La definición de la inversa de una matriz A^{-1} es una matriz tal que $AA^{-1} = I$, donde I es la matriz identidad.

Veamos un ejemplo general para una matriz cuadrada 3 x 3:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} a_{11}' & a_{12}' & a_{13}' \\ a_{21}' & a_{22}' & a_{23}' \\ a_{31}' & a_{32}' & a_{33}' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Se puede plantear como **n** problemas $Ax = b$ con diferentes matrices **b** y la matriz **x** se convierte en la enésima columna de la matriz inversa. Por ejemplo, el primer problema es:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} a_{11}' \\ a_{21}' \\ a_{31}' \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

La segunda columna de la inversa puede calcularse cambiando **b** por $[0,1,0]^T$, la tercera columna por $[0,0,1]^T$, y así sucesivamente. Este método es rápido porque sólo se requiere la sustitución hacia atrás y hacia adelante para resolver los vectores columna después de la descomposición LU inicial.

Es importante destacar que, para obtener la inversa de esta matriz, acabamos teniendo que resolver **n** problemas $Ax = b$ para cada una de las columnas de la inversa. Si se intenta obtener la inversa de la matriz sólo para resolver un problema $Ax = b$, se está introduciendo más errores numéricos en la solución y ralentizando el tiempo de cálculo. Se debe asegurar de que realmente se necesita la inversa de la matriz y nunca utilizar la inversa de la matriz para resolver un sistema de ecuaciones.

5.6. Calcule los factores L y U de la matriz del ejemplo 5.7 y verifique que la factorización LU es inexacta.

Respuesta: Para el desarrollo de este ejercicio, se tomó como base la función definida para el ejercicio b) 3.8, **LU**, que se encarga como su nombre lo dice, de calcular la correspondiente factorización LU de una matriz cuadrada **A** dada, modificando únicamente los valores de retorno que esta función devuelve, es decir, se ha implementado únicamente con el objetivo de retornar la matriz L y la matriz U, sin tomar en cuenta la matriz de permutación P que en la función original (definida en la respuesta de la pregunta b) 3.8) retornaba.

A continuación, se adjuntan las imágenes correspondientes al código fuente de la solución y, además, imagen de los resultados obtenidos al momento de ejecutar el código.

```
function [L, U, P] = LU(A)
%A is the coefficient matrix
%L and U are lu factorized value of A
%P is parmutation matrix
%below is the algorithm for Gaussian elimination
    [m, n] = size(A);
    L = eye(n);
    P = eye(n);
    U = A;
    %initialization of L U and P
    %Loop for LU matrix formation for A matrix
    for k = 1:m-1
        pivot = max(abs(U(k:m,k)));
        for j = k:m
            if(abs(U(j,k)) == pivot)
                ind = j;
                break;
            endif
        endfor
        %updated L U and P values
        U([k,ind],k:m) = U([ind,k],k:m);
        L([k,ind],1:k-1) = L([ind,k],1:k-1);
        P([k,ind],:) = P([ind,k],:);
        %Loop for L and U matrix formation
        for j = k+1:m
            L(j,k) = U(j,k)/U(k,k);
            U(j,k:m) = U(j,k:m)-L(j,k)*U(k,k:m);
        endfor
    endfor
endfunction

function main
  A = [1 1+0.5*(10**-15) 3; 2 2 20; 3 6 4];

  disp("A = ")
  disp(A)
  disp("--------------------")

  [L, U] = LU(A);
  disp("L = ")
  disp(L);
  disp("U = ")
  disp(U);
  disp("--------------------")

  %The accuracy of lu decomposition
  error = norm(L*U-A) / norm(A)

endfunction

main()
```

```
Ventana de comandos
A =
     1.0000     1.0000     3.0000
     2.0000     2.0000    20.0000
     3.0000     6.0000     4.0000
--------------------
L =
   1.0000         0         0
   0.6667    1.0000         0
   0.3333    0.5000    1.0000
U =
     3.0000     6.0000     4.0000
          0    -2.0000    17.3333
          0          0    -7.0000
>>
```

Para comprobar que la descomposición LU obtenida como resultado no es precisa, nos basaremos de la función **norm** que nos ofrece tanto Octave como Matlab para calcular la *Norma Matricial* o simplemente, Norma de una matriz. La cual se define como una medida del tamaño de sus elementos. Es una forma de determinar el "tamaño" de una matriz que no está necesariamente relacionada con el número de filas o columnas que tiene la matriz, de otro modo, la norma de una matriz es un número real que es una medida de la magnitud de la matriz.

De esta forma, para verificar que exactamente la descomposición LU calculada no es precisa, basta con ejecutar la siguiente línea de código luego de haber hecho los cálculos necesarios para dicha descomposición.

```
error = norm(L*U-A) / norm(A)
```

Para la matriz dada, obtendríamos el siguiente resultado para el error de la descomposición LU calculada,

```
--------------------
error = 0.3675
>>
```

Concluyendo que la descomposición realizada, no es precisa para la matriz del ejemplo 5.7 del capítulo.

5.7. Explique por qué el pivoteo parcial por fila no es conveniente para las matrices simétricas.

Respuesta: Una matriz cuya forma cuadrática $x^T Ax$ toma valores positivos y negativos se denomina indefinida. Aunque una matriz indefinida A puede tener una factorización LDL^T, las entradas en estos factores pueden tener una magnitud arbitraria. Es cierto que las estrategias de pivoteo comunes podrían usarse para evitar este inconveniente, pero también es cierto que dichas estrategias destruirían la estructura simétrica de la matriz, y con ello la oportunidad de explotar su estructura en la solución del sistema. Así pues, debemos utilizar un pivoteo simétrico, de la forma $A \leftarrow PAP^T$, pero lamentablemente así no siempre se obtiene estabilidad en la factorización LDL^T.

El reto es entonces utilizar elementos fuera de la diagonal para hacer el pivoteamiento. Con este objetivo surgen dos (2) formas de conseguirlo. La primera forma es mediante el método de Aasen que computa una factorización de la forma PAP^T= LTL^T, donde L es triangular inferior y T es tridiagonal. La segunda forma es utilizando el método de Bunch y Parlett que computa una factorización PAP^T= LDL^T, donde D es una suma directa de bloques pivote de tamaño 1x1 y 2x2 (matriz diagonal por bloques).

Ambas factorizaciones involucran $\frac{n^3}{3}$ flops y una vez computadas pueden utilizarse para resolver el sistema $Ax = b$ en $O(n^2)$.

Como se menciona en un principio, utilizar los métodos comunes de pivotamiento, a pesar de que podamos utilizarlas para matrices simétricas, existe la posibilidad de destruir la simetría de dichas matrices y de esta forma, perder ciertas ventajas que estas nos puede llegar a ofrecer. También, una forma de realizar una estrategia de pivotamiento, es basarnos de un pivotamiento diagonal para matrices simétricas indefinidas.

5.8.

Respuesta: Consideremos la siguiente matriz cuadrada 3x3 **A** en su representación general,

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Además, sabemos que, a través de la factorización LU, obtendremos las siguientes matrices, también consideradas en su representación general, tomando la matriz L como triangular inferior con 1´s en su diagonal principal.

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix}; U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Por último, sabemos que $A = LU$, teniendo así,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ u_{11}l_{31} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{pmatrix}$$

$$A = \begin{pmatrix} 2 & -2 & 0 \\ \varepsilon - 2 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix},$$

Donde,

$$-u_{11} = 2; u_{12} = -2; u_{13} = 0$$

$$l_{21}u_{11} = \varepsilon - 2 \Rightarrow l_{21} = \frac{\varepsilon - 2}{2};$$

$$l_{21}u_{12} + u_{22} = 2 \Rightarrow u_{22} = 2 + \varepsilon - 2 = \varepsilon;$$

$$u_{11}l_{31} = 0 \Rightarrow l_{31} = 0;$$

$$l_{31}u_{12} + l_{32}u_{22} = -1 \Rightarrow l_{32} = \frac{-1}{\varepsilon};$$

$$l_{31}u_{13} + l_{32}u_{23} + u_{33} = 3 \Rightarrow u_{33} = 3;$$

$$u_{23} = 0$$

Teniendo así,

$$L=\begin{pmatrix}1 & 0 & 0\\ \dfrac{\varepsilon-2}{2} & 1 & 0\\ 0 & \dfrac{-1}{\varepsilon} & 0\end{pmatrix};U=\begin{pmatrix}2 & -2 & 0\\ 1 & \varepsilon & 0\\ 0 & 0 & 3\end{pmatrix}$$

Considerando que,

$$l_{32}=\frac{-1}{\varepsilon};\ l_{32}\to\infty,\varepsilon\to 0$$

Si tomamos,

$$\delta=\begin{pmatrix}C_1\\ C_2\\ C_3\end{pmatrix};entonces,$$

$$\begin{pmatrix}1 & 0 & 0\\ \dfrac{\varepsilon-2}{2} & 1 & 0\\ 0 & \dfrac{-1}{\varepsilon} & 1\end{pmatrix}\begin{pmatrix}C_1\\ C_2\\ C_3\end{pmatrix}=\begin{pmatrix}0\\ \varepsilon\\ 2\end{pmatrix}$$

Y si consideramos,

$$C=\begin{pmatrix}C_1\\ C_2\\ C_3\end{pmatrix}=\begin{pmatrix}0\\ \varepsilon\\ 3\end{pmatrix};hacemos\ Ux=C,tendremos,$$

$$\begin{pmatrix}2 & -2 & 0\\ 0 & \varepsilon & 0\\ 0 & 0 & 3\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\begin{pmatrix}0\\ \varepsilon\\ 3\end{pmatrix}\Rightarrow\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\begin{pmatrix}1\\ 1\\ 1\end{pmatrix}$$

Quedando demostrado que el sistema de ecuaciones $Ax=b$, no se verá afectado por el valor de ε.

5.9.

Respuesta: Para la resolución del ejercicio, nos hemos basado en las funciones implementadas para el cálculo de la descomposición LU como la sustitución hacías atrás como sustitución hacía adelante aplicadas en los ejercicios b) 3.8 y c) 5.6, para hallar la solución de un Sistema de Ecuaciones Lineales a partir de estos métodos.

La diferencia principal es que para cada variación de la matriz inicial **A**, se indica el error que tiene la solución calculada en comparación al verdadero valor de la solución, es decir, por cada vector **x** hallado, observaremos que al aplicar el método indicado (Factorización Gaussiana usando el pivotamiento parcial por filas, para este caso, basándonos de la Factorización LU), este arrojará resultados no precisos en comparación a la verdadera solución.

Por ello, además de indicar por impresión en pantalla el vector **x**, también se indica el error obtenido haciendo uso de la *Norma Matricial* o simplemente *Norma de una matriz*, de la cual hemos mencionado anteriormente en otras respuestas de esta sección de la asignación.

Recordemos que la *Norma Matricial* se define como una medida del tamaño de sus elementos. Es una forma de determinar el "tamaño" de una matriz que no está necesariamente relacionada con el número de filas o columnas que tiene la matriz, de otro modo, la norma de una matriz es un número real que es una medida de la magnitud de la matriz, pero en este caso, calcularemos la norma de,

$$error = norm(X - X2),$$

Donde,

$$X\ es\ igual\ a\ la\ solución\ obtenida\ y\ X2 = [1 \quad 1 \quad 1 \quad 1]^T$$

Esto, para cada matriz,

$$A_1; A_2; A_3$$

Obteniendo así, los siguientes resultados para cada variación del Sistema de Ecuaciones Lineales,

- Para la matriz A_1:

```
Ventana de comandos
The A1 matrix is
   15     6     8    11
    6     6     5     3
    8     5     7     6
   13     3     6     9
The b matrix is
   40
   20
   26
   31
The solution matrix X is
   1.0000
   1.0000
   1.0000
   1.0000
Error in solution of x is 6.048423e-15
--------------------
```

- Para la matriz A_2:

```
--------------------
The A2 matrix is
   468   199   272   330
   205   106   131   141
   284   131   174   199
   378   153   215   269
The b matrix is
   1269
    583
    788
   1015
The solution matrix X is
   1.0000
   1.0000
   1.0000
   1.0000
Error in solution of x is 8.664848e-14
--------------------
```

- Para la matriz A_3:

```
--------------------
The A3 matrix is
   4.5734e+08   1.9851e+08   2.6908e+08   3.2223e+08
   2.0631e+08   8.9557e+07   1.2139e+08   1.4536e+08
   2.8174e+08   1.2229e+08   1.6577e+08   1.9851e+08
   3.6736e+08   1.5945e+08   2.1614e+08   2.5884e+08
The b matrix is
   1.2472e+09
   5.6261e+08
   7.6831e+08
   1.0018e+09
The solution matrix X is
   1.0000
   1.0000
   1.0000
   1.0000
Error in solution of x is 7.618621e-08
>>
```

Observando así que cada solución no es precisa, sino que existe un “margen” de error en comparación con la verdadera solución del Sistema de Ecuaciones.