UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

LABORATORIO 1 ALGORITMOS NUMÉRICOS: MÉTODO NEWTON MULTIVARIABLE Y MÉTODOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES

GUSTAVO HURTADO

Profesor: Óscar Rojas Díaz.

Santiago - Chile

14 de mayo de 2019

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

CAPÍTULO 1. INTRODUCCIÓN

Si bien existe una manera análoga de resolver sistemas de ecuaciones lineales o no lineales, en la realidad, es una tarea prácticamente imposible de realizar cuando se piensa en miles o cientos de miles de datos. Aquí radica la importancia de los métodos numéricos, ya que permiten trabajar con grandes cantidades de datos y encontrar soluciones a estos sistemas, que si bien pueden no ser exactas, nos entregan una muy buena aproximación de los resultados.

En el siguiente informe se hará un análisis al método para resolver sistemas de ecuaciones no lineales llamado *Newton Rapson Multivariable*. Además, se analizarán 7 métodos de resolución de sistemas de ecuaciones lineales, los cuales se pueden dividir en 2 tipos: directos e iterativos. Dentro de los métodos directos se encuentran *QR , Cholesky, Doolittle, Givens y Householder*, mientras que en los directos están *Gauss-Jacobi y Gauss-Seidel.*

Por otro lado, se desarrollará un gestor que permita discriminar según el tipo y tamaño de la matriz de entrada, qué método de resolución de ecuaciones lineales utilizar, analizando propiedades tales como simetría, diagonal dominante, dispersión, etc. De este gestor se realizará un análisis de eficiencia, para verificar si es más conveniente hacer uso de él, o simplemente ejecutar todos los métodos y escoger alguno.

CAPÍTULO 2. MARCO TEÓRICO

En este capítulo se presenta una breve descripción de cada uno de los métodos.

2.1 SISTEMA DE ECUACIONES NO LINEALES

2.1.1. Newton Multivariable

Se considera la ecuación $f(x) = 0$ (donde f es una función vectorial de variable vectorial) equivalente a un sistema de ecuaciones no lineales.

Sea x la solución exacta del sistema y x_n una aproximación de ella. Si se denota por $h = x - x_n$ se tiene que $x = x_n + h$, por lo que luego de hacer series de Taylor, se obtiene una aproximación del tipo:

$$x_{n+1} = x_n - f'^{-1}(x_n)f(x_n)$$

Por lo que en cada iteración de éste método, se reduce al cálculo del vector h correspondiente y éste no es más que la solución del sistema de ecuaciones lineales:

$$f'(x_n)h + f(x_n) = 0$$

Para este caso en específico, se hará uso de las funciones:

1. $x_1^2 + x_2 - 37 = 0$.
2. $x_1 + x_2^2 - 5 = 0$
3. $x_1 + x_2 + x_3 - 3 = 0$

Usando como vector inicial $X_{(0)} = (0, 0, 0)^T$

2.2 SISTEMAS DE ECUACIONES LINEALES

2.2.1. Métodos Iterados

Un método iterado de resolución del sistema $Ax = b$ es aquel que genera, a partir de un vector inicial x_0, una sucesión de vectores $x1, x2, ...$ El método se dirá que es consistente con el sistema $Ax = b$, si el límite de dicha sucesión, en caso de existir, es solución del sistema. Se dirá que el método es convergente si la sucesión generada por cualquier vector inicial x_0 es convergente a la solución del sistema.

Es evidente que si un método es convergente es consistente, sin embargo, el recíproco no es cierto.

1. **Gauss Jacobi**: Se puede apreciar en la *Figura 2-1* la manera en que Gauss Jacobi realiza el cálculo de cada uno de los x_i, haciendo uso de una matriz A y b.

$$x_i^{(s+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(s)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(s)}}{a_{ii}} \qquad i = 1 : n$$

Figura 2-1: Cálculo de Gauss Jacobi.

2. **Gauss Seidel**: Si bien Gauss Seidel es prácticamente similar a Jacobi, en este caso, las componentes ya calculadas se usan para calcular las siguientes en la misma iteración, Esto acelera la convergencia, sin embargo no se puede asegurar que en general sea mejor que Jacobi.

$$x_i^{(s+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(s+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(s)}}{a_{ii}} \qquad i = 1 : n$$

Figura 2-2: Cálculo de Gauss Seidel.

2.2.2. Métodos Directos

Se dice que un método para resolver un problema es directo si tras un número finito de operaciones con los datos del problema se obtiene una solución al problema, que, en general, será aproximada.

El fundamento básico de estos métodos es que existen sistemas fáciles de resolver, así que se intentará transformar el problema que se tiene a uno fácil de resolver.

1. **Doolittle**: Se basa en la factorización LU de la matriz del sistema, de forma que quede el producto de una matriz triangular superior (U) y una triangular inferior (L):

$$A = LU = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}$$

Figura 2-3: Descomposición LU.

En principio, al operarla saldría n^2 ecuaciones con $n^2 + 1$ incógnitas, lo cual no es nada práctico. Para evitarlo, se igualan los elementos de la diagonal principal de la matriz L triangular a uno $(lii = 1)$.

$$u_{1i} = a_{1i} \qquad i = 1 : n \qquad l_{11} = 1 \qquad l_{i1} = \frac{a_{i1}}{u_{11}} \qquad i = 2 : n$$

$$j = 2 : n \begin{cases} u_{ji} = a_{ji} - \sum_{k=1}^{j-1} l_{jk} u_{ki} & i = j : n \\ l_{jj} = 1; \ \text{Si } j < n : \ l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) / u_{jj} & i = j+1 : n \end{cases}$$

Figura 2-4: Cálculo de la factorización mediante Doolittle.

.

2. **Choleskyl**: Se utiliza cuando la matriz del sistema es simétrica: $A = A^t$

$$A = LL^t$$

$$A = \begin{pmatrix} l_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & \dots & l_{n1} \\ \vdots & \ddots & \vdots \\ 0 & \dots & l_{nn} \end{pmatrix}$$

Figura 2-5: Desconposición LL^t.

En la *Figura 2-6* se puede apreciar cómo se realiza la factorización Cholesky.

$$l_{11} = \sqrt{a_{11}} \qquad l_{i1} = \frac{a_{i1}}{l_{11}} \qquad i = 2:n$$

$$j = 2:n \begin{cases} l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2} \\ \text{Si } j < n; \quad l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) / l_{jj} \qquad i = j+1:n \end{cases}$$

Figura 2-6: Cálculo de la factorización mediante Cholesky.

3. **Givens**: Consiste en sucesivas transformaciones de rotación usando matrices de Givens (Ver *Figura 2-7*) hasta reducir la matriz A rectangular *m* * *n* a una triangular. De esta forma, es posible lograr obtener una factorización QR.

$$G_{pq} = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & \cos\theta & \dots & \operatorname{sen}\theta & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & -\operatorname{sen}\theta & \dots & \cos\theta & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \begin{matrix} \\ \\ p \\ \\ q \\ \\ \\ \end{matrix}$$

$$\qquad\qquad\qquad\quad p \qquad\quad q$$

Figura 2-7: Matriz Givens.

Todos los pasos que se dan están asociados a transformaciones ortogonales, sin embargo, resulta costoso ya que cada rotación hace un único cero en la matriz A, por lo que para una matriz de orden n serían necesarias $(n^2 - n)/2$ rotaciones.

4. **HouseHolder**: Otra posibilidad es hacer reflexiones en vez de rotaciones, ya que estas consiguen anular todos los elementos situados por debajo de uno prefijado de una determinada columna.

   Una transformación de Householder o reflexión de Householder es una transformación que refleja el espacio con respecto a un plano determinado. Esta propiedad se puede utilizar para realizar la transformación QR de una matriz si tenemos en cuenta que es posible elegir la matriz de Householder de manera que un vector elegido quede con una única componente no nula tras ser transformado (es decir, premultiplicando por la matriz de Householder). Gráficamente, esto significa que es posible reflejar el vector elegido respecto de un plano de forma que el reflejo quede sobre uno de los ejes de la base cartesiana.

   De esta manera se puede obtener la transformación QR de manera más rápida, aunque, con una mayor pérdida de precisión que con Givens.

5. **QR**: Dada una matriz A (no necesariamente cuadrada), con columnas linealmente independientes, se encontrarán matrices Q, R tales que:

   a) A = QR.

   b) Las columnas de Q son ortonormales.

   c) Q es del mismo tamaño que A.

   d) R es triangular superior invertible.

   La forma de hacerlo es aplicar el proceso de Gram-Schmidt a las columnas de A, como se puede apreciar en la *Figura 2-8*.

Proceso de Gram Schmidt.
A partir de los vectores linealmente independientes $\mathbf{v}_1, \ldots, \mathbf{v}_n$ se construyen

$$\mathbf{u}_1 = \mathbf{v}_1, \qquad \mathbf{u}_k = \mathbf{v}_k - \sum_{j=1}^{k-1} \frac{\langle \mathbf{v}_k, \mathbf{u}_j \rangle}{\|\mathbf{u}_j\|^2} \mathbf{u}_j, \ j = 2, \ldots, k.$$

Los vectores $\mathbf{u}_1, \ldots, \mathbf{u}_n$ son ortogonales.

Figura 2-8: Proceso de Gram Schmidt.

2.3 GESTOR DE MÉTODOS PARA SISTEMAS DE ECUACIONES LINEALES

Para efectos de este laboratorio se solicitaba generar un gestor para los diferentes métodos desarrollados, teniendo en consideración propiedades de las matrices entrantes tales como esparcimiento, simetría, definida positiva, entre otras y haciendo uso de esa información, decidir qué método es mejor para ser utilizado.

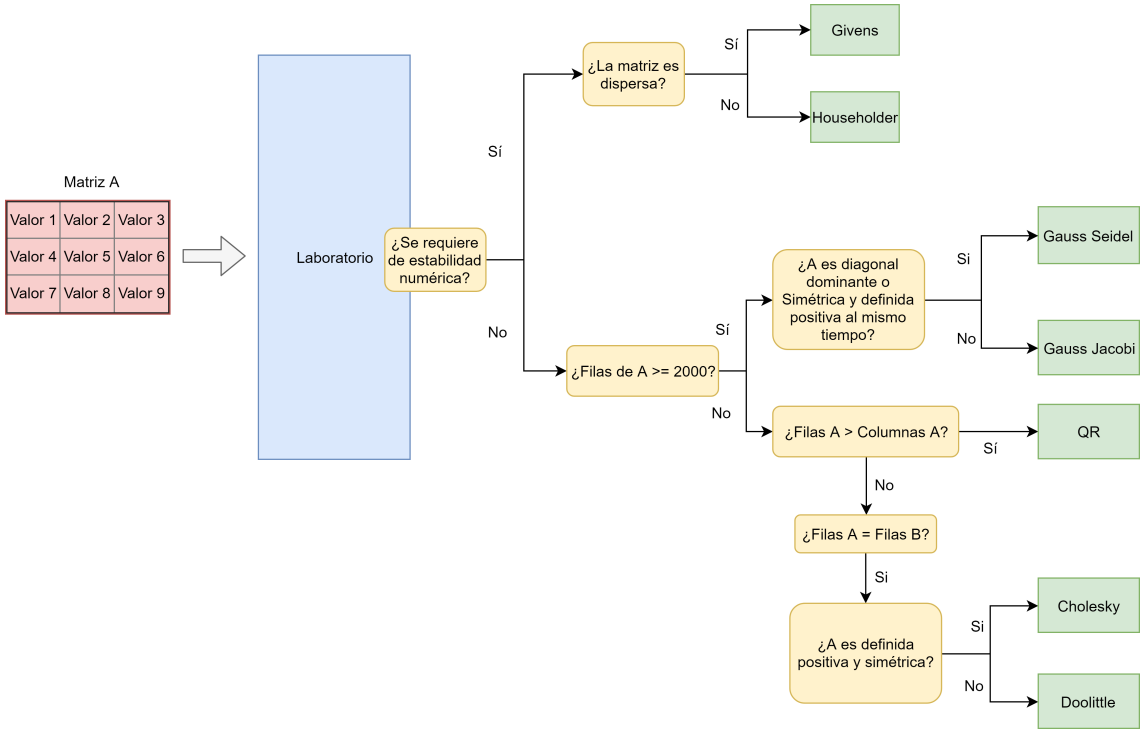

Figura 2-9: Gestor de métodos

Los elementos que se tomaron en consideración para realizar este gestor fueron en primer lugar, si se requiere de una estabilidad numérica, es decir, un bajo error. Para este propósito, los métodos de Givens y Householder son los ideales. Por otro lado, se consideró el tamaño de la matriz entrante. En caso de que sean matrices de gran tamaño, Gauss - Seidel y Gauss - Jacobi son los métodos por excelencia, debido a la flexibilidad de su eficiencia y eficacia que varía según la cantidad de iteraciones que se le asigna, por lo que es posible manejar el error.

Por otro lado se tiene que si la cantidad de filas es mayor a la cantidad de columnas, el método directo QR permite realizar aproximaciones mucho más eficaces que los otros métodos.

Finalmente, en caso de que la cantidad de filas sea igual a la de columnas, es decir, se tiene una matriz cuadrada, se puede elegir entre el método de Cholesky o Doolittle, dependiendo de si la matriz es una definida positiva y a la vez es simétrica.

Para generar este gestor se hizo uso del material facilitado en el moodle del curso, habiendo sido .ªpuntes de Métodos Numéricos 2do E.T.S.I. Telecomunicación Universidad de Málaga", la principal fuente de información para establecer los criterios del gestor.

CAPÍTULO 3. RESULTADOS

3.1 NEWTON MULTIVARIABLE

En esta sección se mostrarán los resultados del método de resolución de sistemas no lineales *Newton Multivariable*.

Como antes se mencionó, las funciones y el vector inicial que fueron utilizandos para este método son:

1. $x_1^2 + x_2 - 37 = 0$.
2. $x_1 + x_2^2 - 5 = 0$
3. $x_1 + x_2 + x_3 - 3 = 0$

$X_{(0)} = (0, 0, 0)^T$

Luego de realizar varias pruebas, se encontró que con una máximo de 12 iteraciones, se llega al error mínimo, que en este caso, es 0, es decir, no existe error en el cálculo de las variables.

A continuación se mostrarán los resultados obtenidos en estas pruebas, mostrando tanto los valores de las variables X_1, X_2, X_3, el de las funciones evaluadas en las funciones antes mencionadas y finalmente el error por cada iteración del método.

1. **Valores de las variables:**

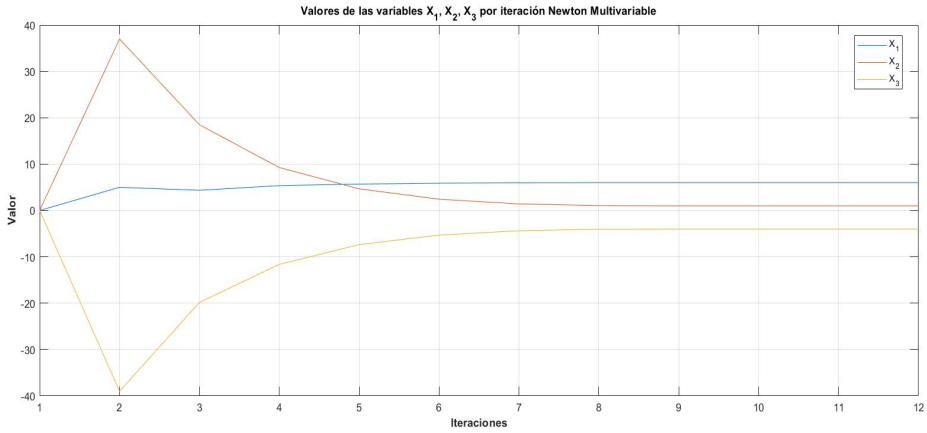

Figura 3-1: Valores de X_1, X_2, X_3.

Se puede apreciar en la *Figura 3-1* como a medida que van ocurriendo las iteraciones, las variables convergen a un valor, los cuales son: $x_1 = 6$, $x_2 = 1$, $x_3 = -4$.

2. **Valores de las funciones evaluadas en X_1, X_2, X_3:**

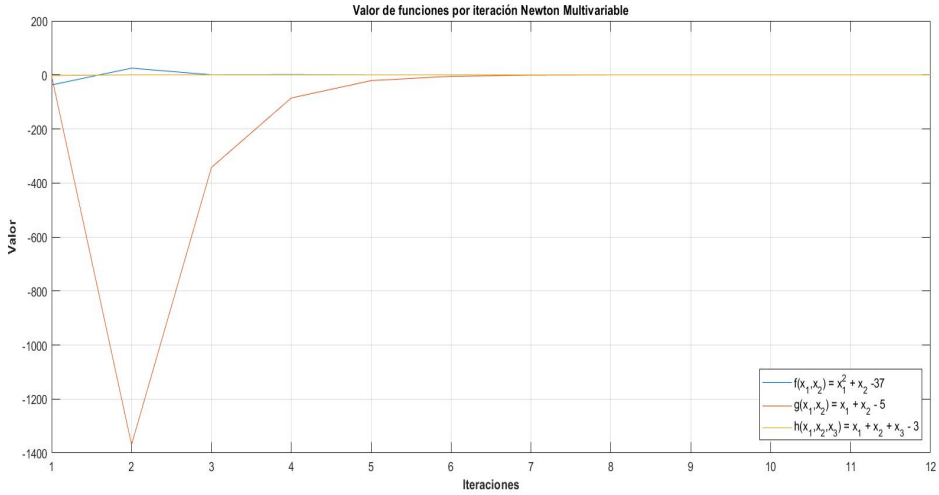

Figura 3-2: Valores de cada función

Lo mismo ocurre con el valor de las funciones. A medida que que las variables X_1, X_2, X_3 van convergiendo a su valor, las funciones convergen a 0.

3. **Valores de los errores:**

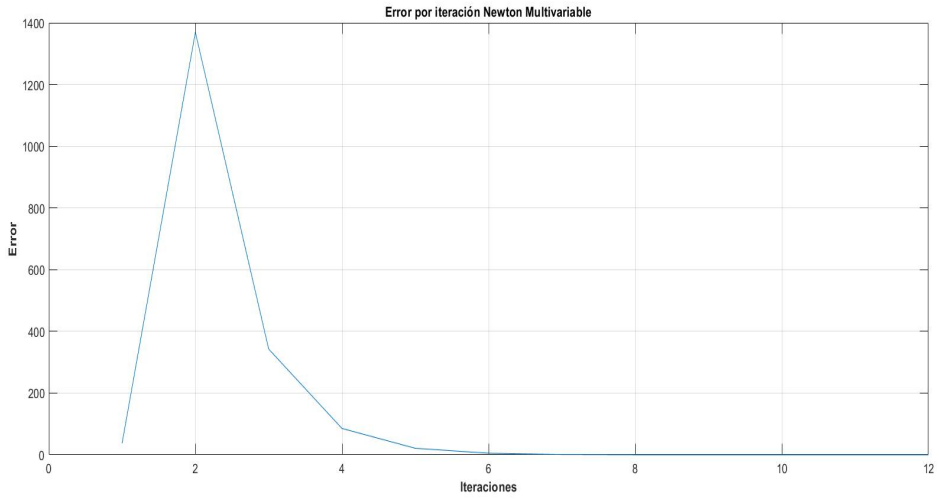

Figura 3-3: Valores de los errores

En la *Figura 3-3* se puede apreciar como el error que en un comienzo es bastante alto, va disminuyendo de forma exponencial hasta converger poco a poco al valor 0.

3.2 GAUSS - JACOBI VS GAUSS - SEIDEL

Antes de mostrar los resultados obtenidos para todos los métodos, se hará una muestra con respecto a la convergencia que tienen estos métodos iterativos.

Cabe mencionar que se realizaron pruebas para matrices de tamaño 289x289, 1089x1089 y 4225x4225, haciendo uso de 100 iteraciones para ambos métodos.

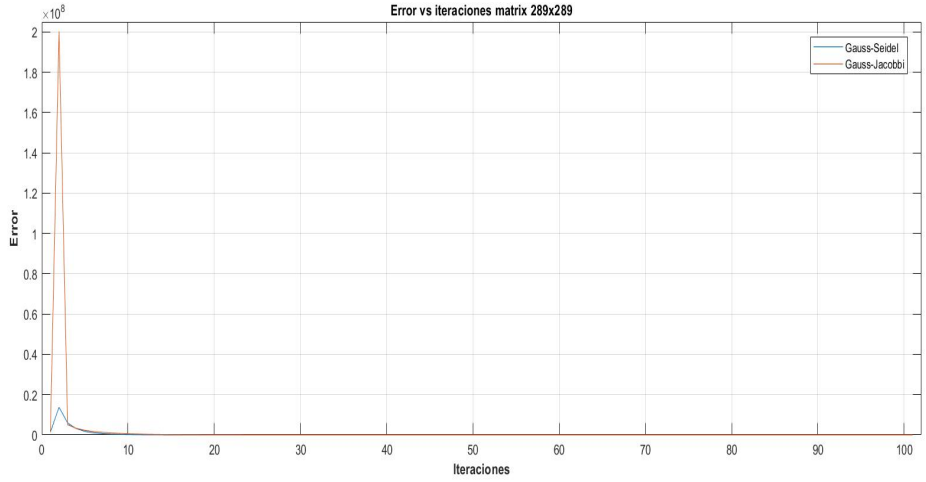

Figura 3-4: Valores de errores en matríz 289x289

A simple vista en la *Figura 3-4* es difícil ver cómo se comportan los errores de estos métodos, pero si aplicamos un logaritmo al eje Y, obtendremos lo siguiente:

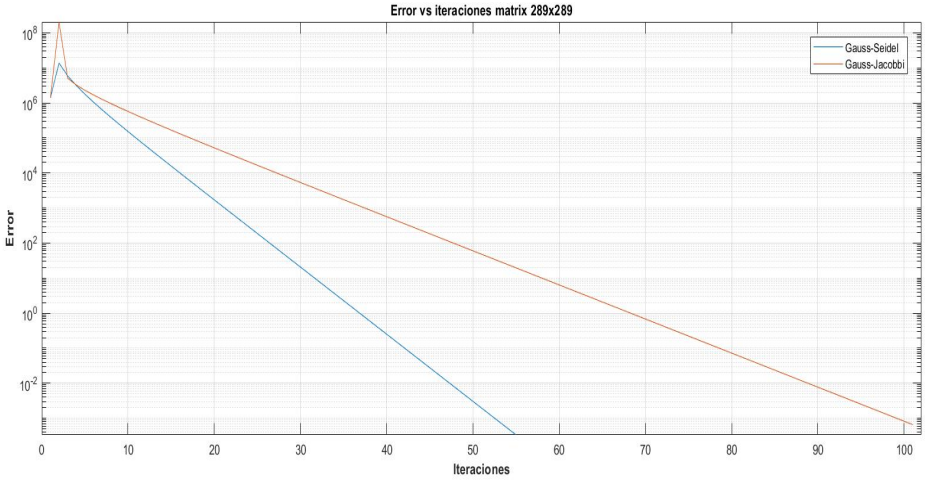

Figura 3-5: Valores de errores en matríz 289x289 con eje Y log

En la *Figura 3-5* se puede apreciar cómo Gauss-Seidel y Gauss-Jacobi convergen de manera exponencial, ya que lo que se ve al aplicar logaritmo, es una recta. Ahora, es notoria la diferencia en sus convergencias, ya que Gauss-Seidel entre la iteración 50 y 60 llega a la misma tolerancia a la que Gauss-Jacobi lo hacen en la iteración 100.

Este mismo patrón se repite en las otras matrices, que se puede apreciar en las *Figuras 3-6,3-7* para la matriz de 1089x1089, y en las *Figuras 3-8,3-9* para la matriz de 4225x4225.

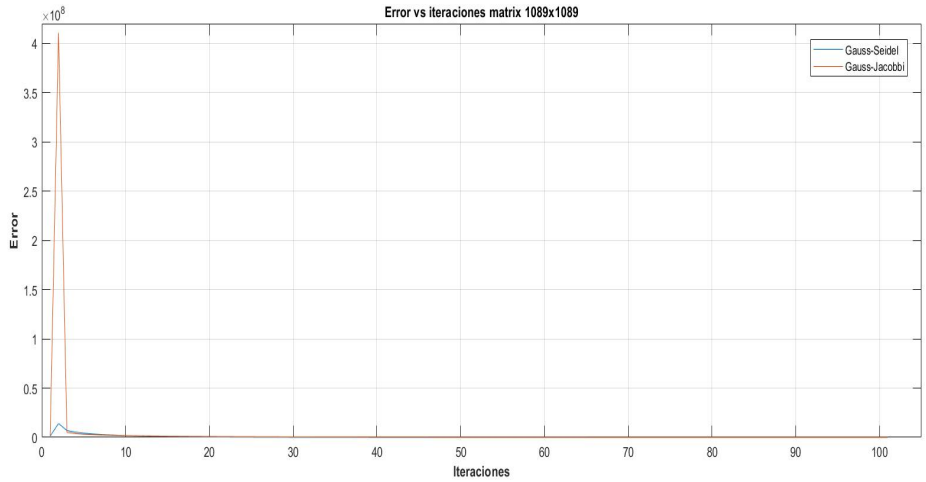

Figura 3-6: Valores de errores en matríz 1089x1089

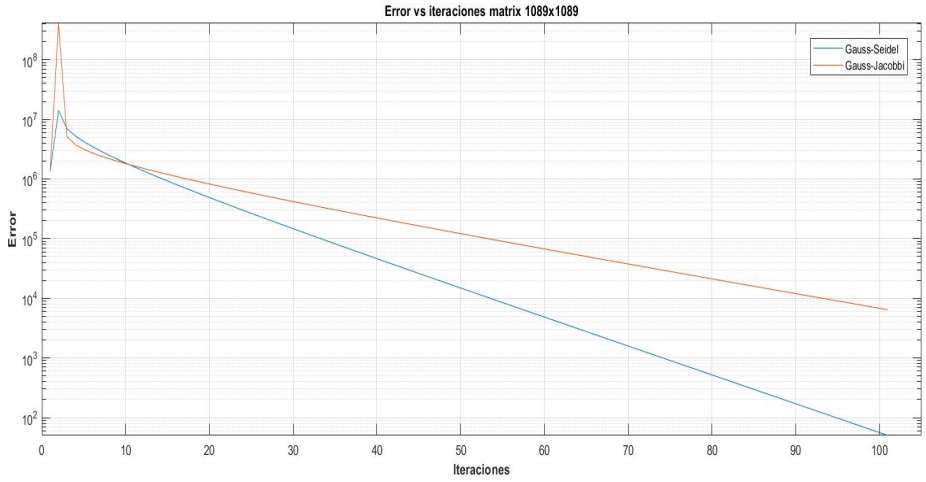

Figura 3-7: Valores de errores en matríz 1089x1089 con eje Y log

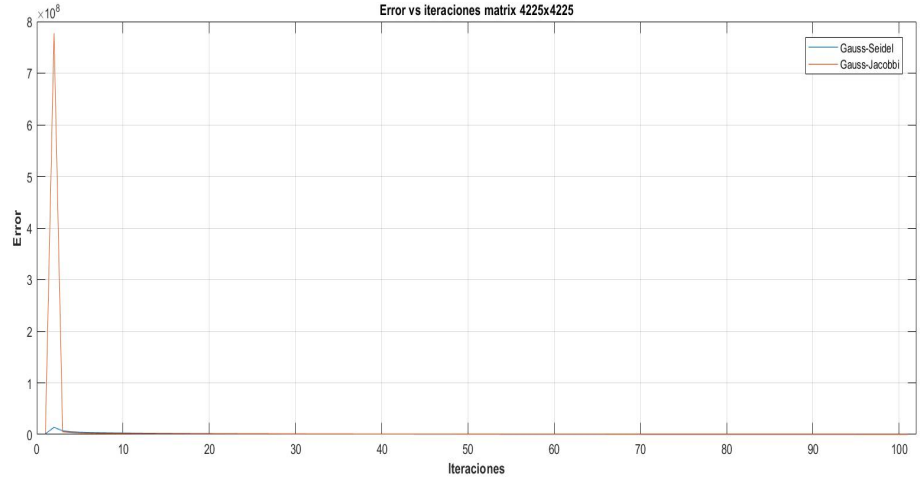

Figura 3-8: Valores de errores en matríz 4225x4225

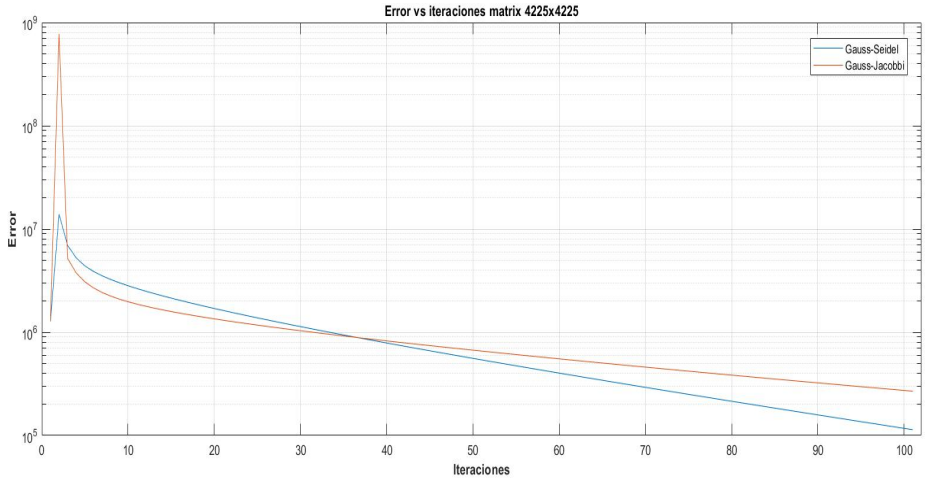

Figura 3-9: Valores de errores en matríz 4225x4225 con eje Y log

3.3 MÉTODOS ITERATIVOS Y MÉTODOS DIRECTOS

En la siguiente sección se mostrarán los resultados obtenidos con los métodos iterativos de Gauss-Jacobi, Gauss-Seidel y los métodos directos Givens, Householder, Doolittle, QR y Cholesky.

Para estos 7 métodos se mostrarán su error, tiempos de ejecución y costos operacionales, y al igual que en la sección anterior, se mostrarán los resultados para las matrices de 289x289, 1089x1089 y 4225x4225, tanto como los valores del eje Y linean como con log, para darle una mejor interpretación a los datos obtenidos. Se obviarán los resultados de las soluciones, ya que gráficamente los 7 métodos figuran con exactamente los mismos resultados, mientras que numéricamente, las diferencias son del orden de 10^{-5} aproximadamente.

En el caso de los métodos iterativos Gauss-Jacobi y Gauss-Seidel se hizo uso de una cantidad de iteraciones variables según la matriz de entrada. Es decir, que para matrices con una cantidad de filas menores a 1000, se realizan 100 iteraciones, para una cantidad de filas entre 1000 y 4000, son 500 iteraciones, mientras que para el resto de matrices, se realizan 1000 iteraciones. Este ajuste se realizó para que la comparación de los métodos sea justa, ya que si se mantenía una baja cantidad de iteraciones, el error se disparaba, mientras que su costo operacional y tiempo de ejecución eran bastante bajos, pero si se dejaba una muy alta cantidad de iteraciones (más de 1000), el error era bastante bajo, pero su tiempo de ejecución y costos operacionales se disparaba, así que de esta manera se ajustó para que haya un equilibrio entre eficiencia y efectividad.

Es importante mencionar que para la matriz de 4225x4225 se descartó totalmente el uso de Givens debido a que su ejecución luego de 3 días no terminaba, por lo que todos los valores en los gráficos de esta matriz, se visualizarán con 0.

Por otro lado, para el cálculo del costo operacional, se hizo la suma de las operaciones que ocurren dentro cada método. Esta suma de operaciones toma por igual la suma, resta, igualación, multiplicación, etc. Esto se aplica también para la suma o multiplicación de matrices.

3.3.1. Errores obtenidos

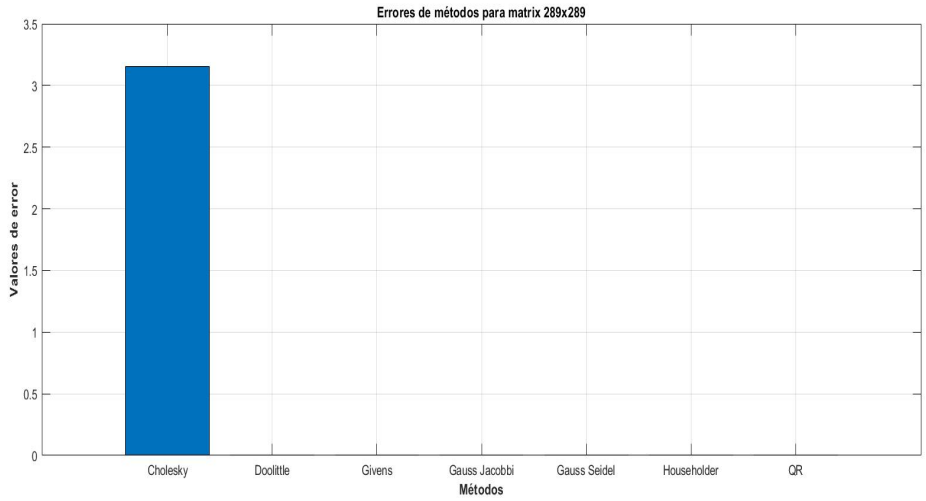

Figura 3-10: Valores de errores en matríz 289x289

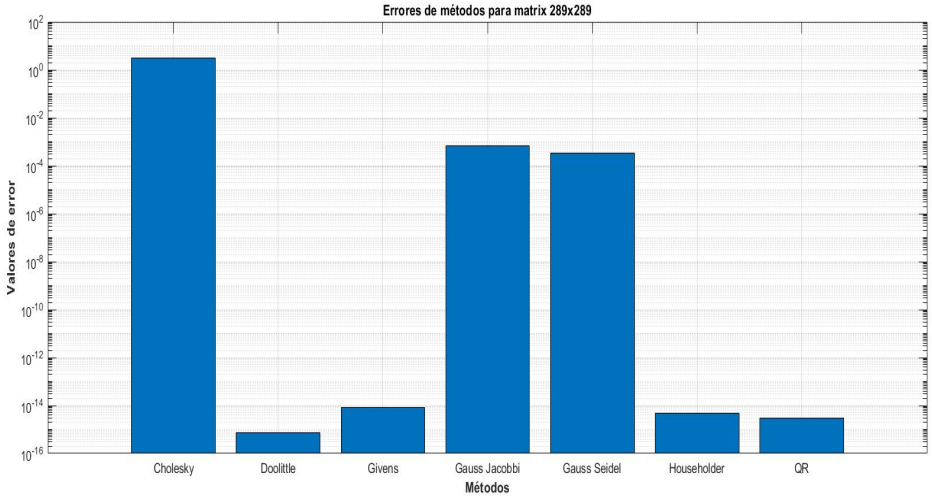

Figura 3-11: Valores de errores en matríz 289x289 con Y log

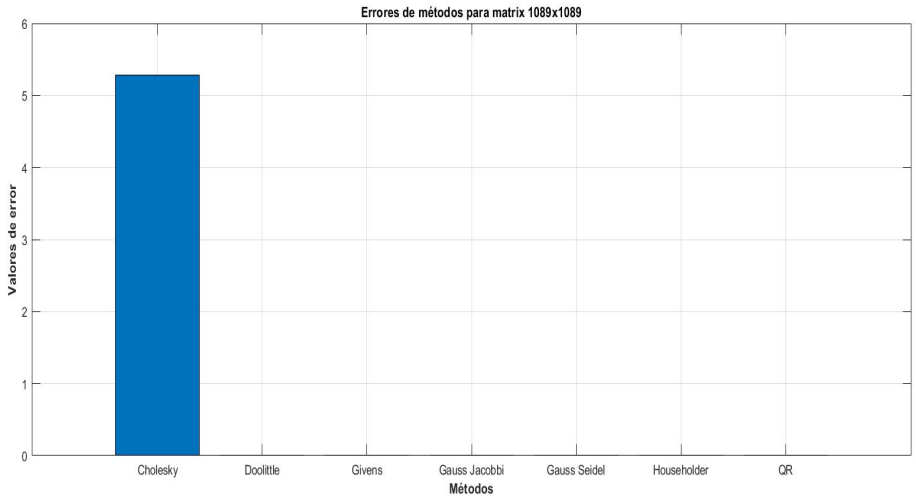

Figura 3-12: Valores de errores en matríz 1089x1089

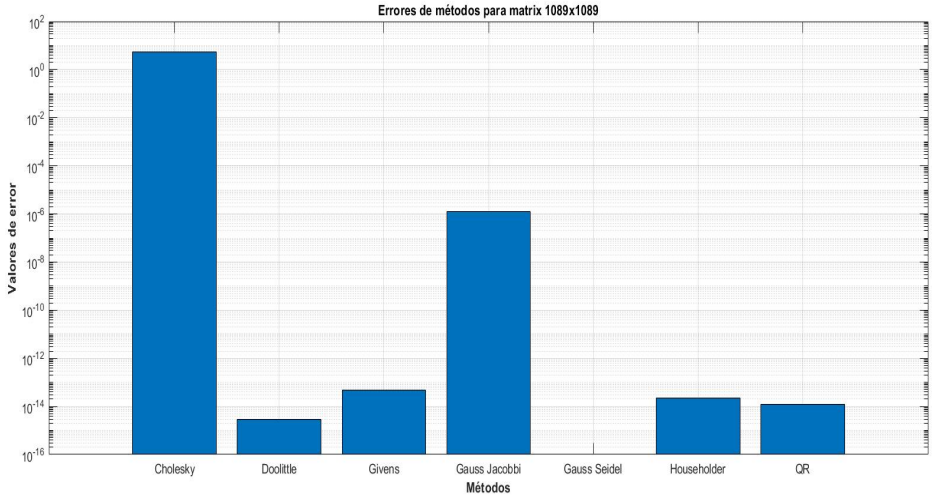

Figura 3-13: Valores de errores en matríz 1089x1089 con Y log

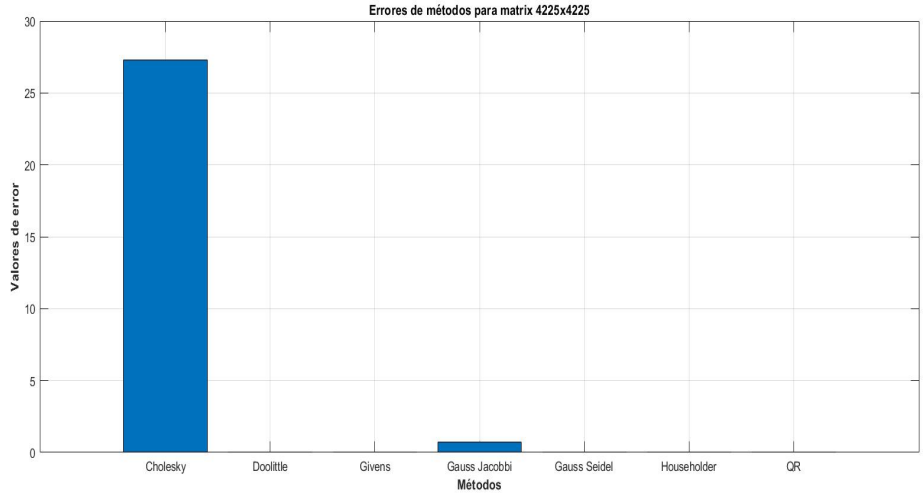

Figura 3-14: Valores de errores en matríz 4225x4225

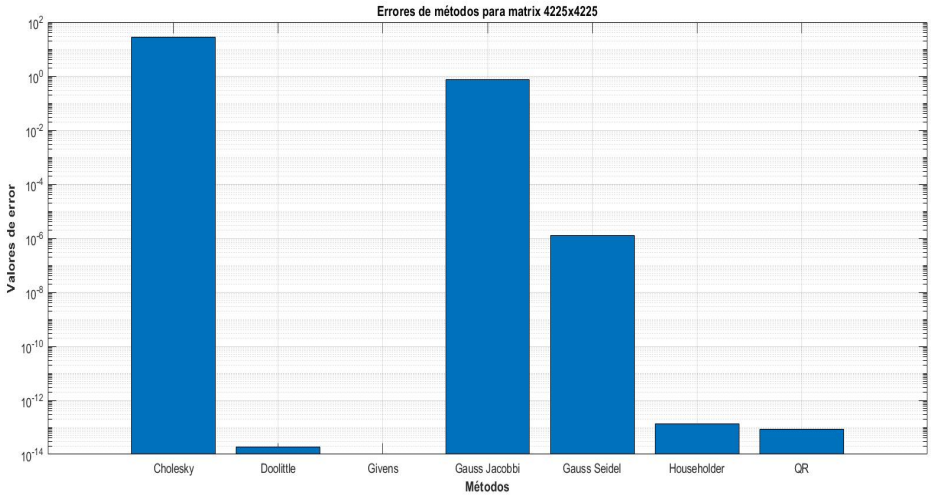

Figura 3-15: Valores de errores en matríz 4225x4225 con Y log

3.3.2. Tiempos de ejecución obtenidos

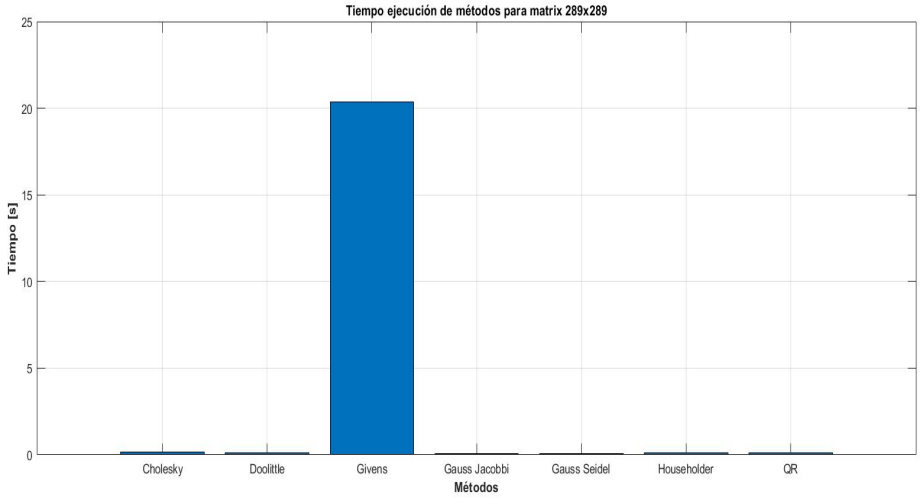

Figura 3-16: Valores de tiempos de ejecución en matríz 289x289

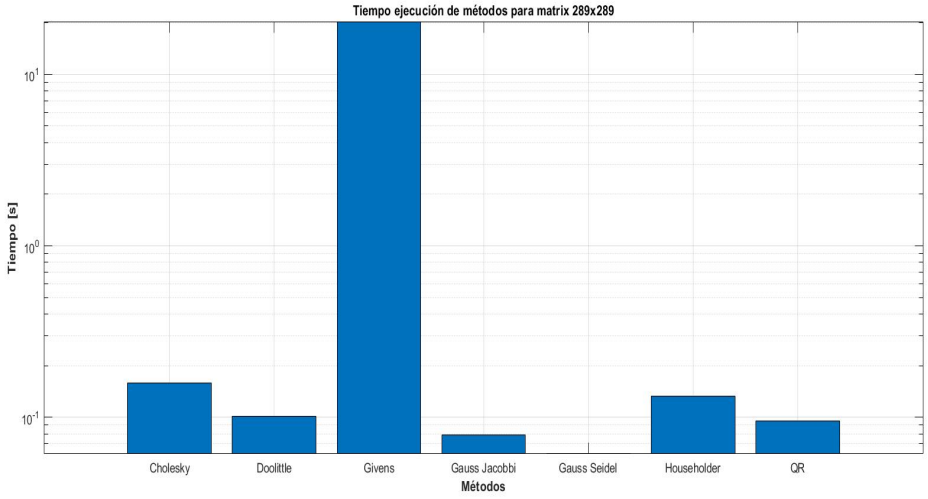

Figura 3-17: Valores de tiempos de ejecución en matríz 289x289 con Y log

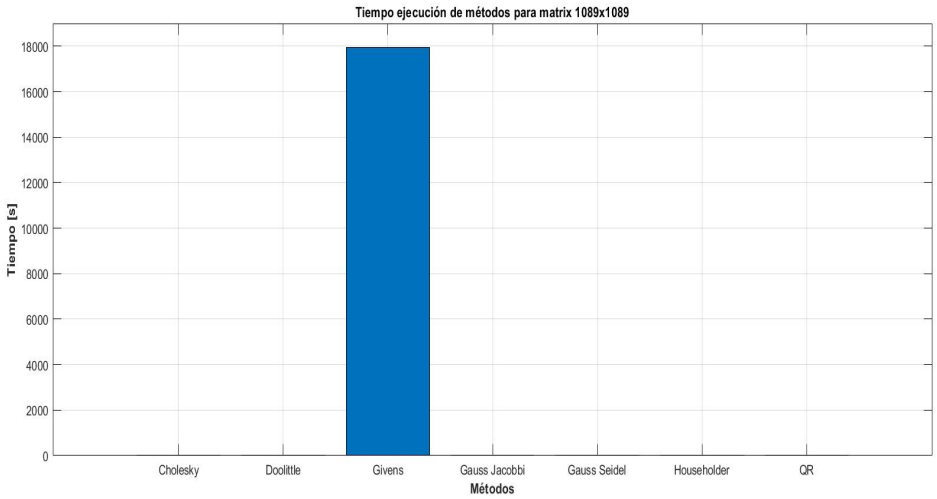

Figura 3-18: Valores de tiempos de ejecución en matríz 1089x1089

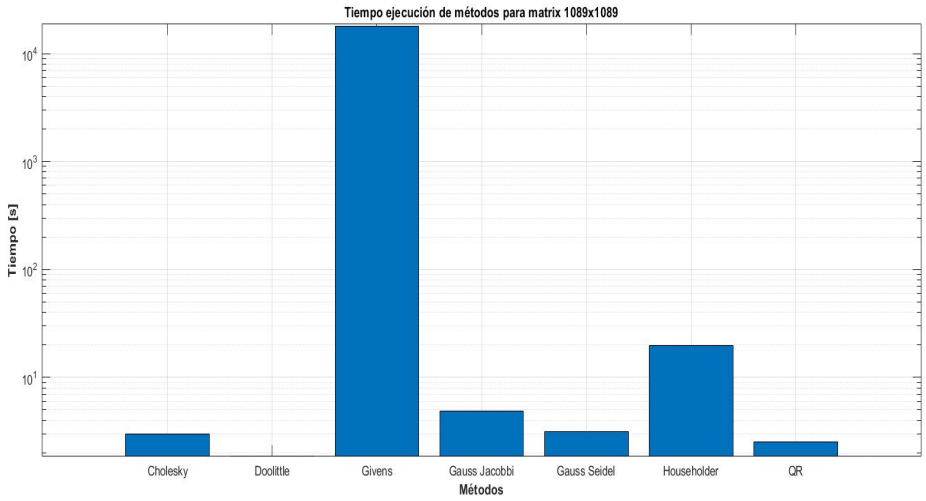

Figura 3-19: Valores de tiempos de ejecución en matríz 1089x1089 con Y log

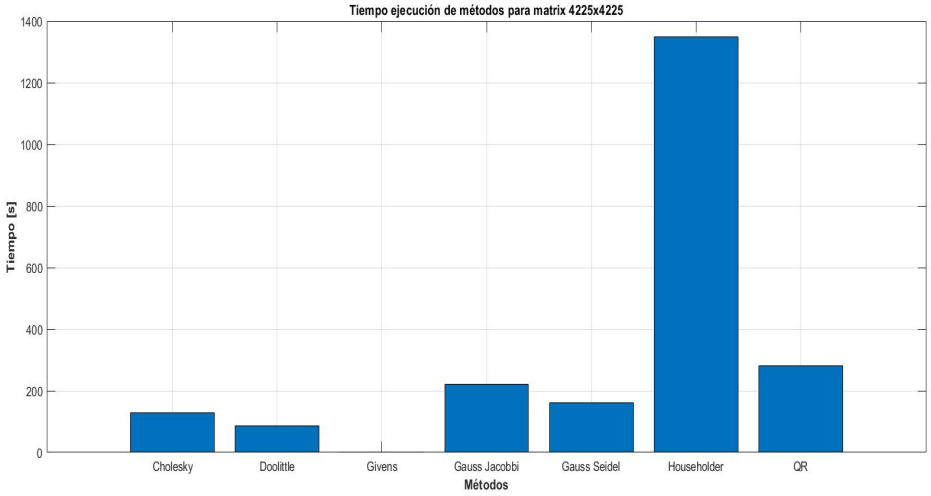

Figura 3-20: Valores de tiempos de ejecución en matríz 4225x4225

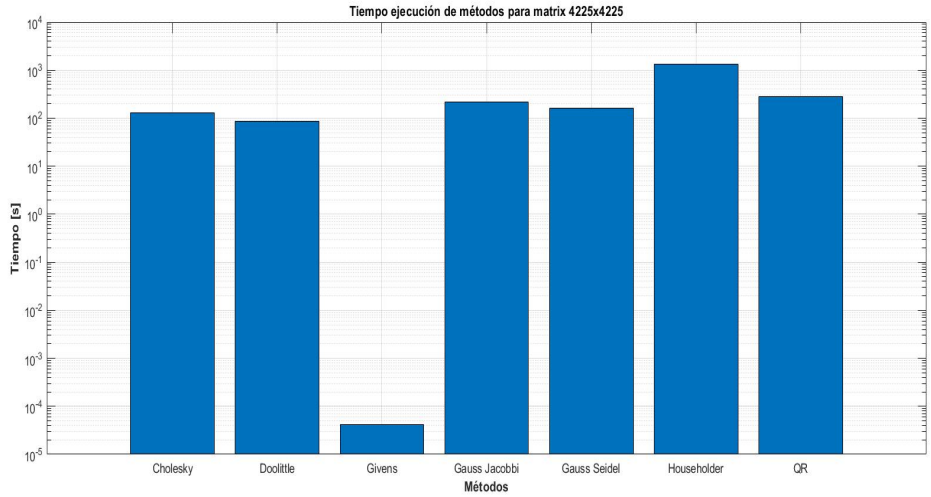

Figura 3-21: Valores de tiempos de ejecución en matríz 4225x4225 con Y log

3.3.3. Costos operacionales obtenidos

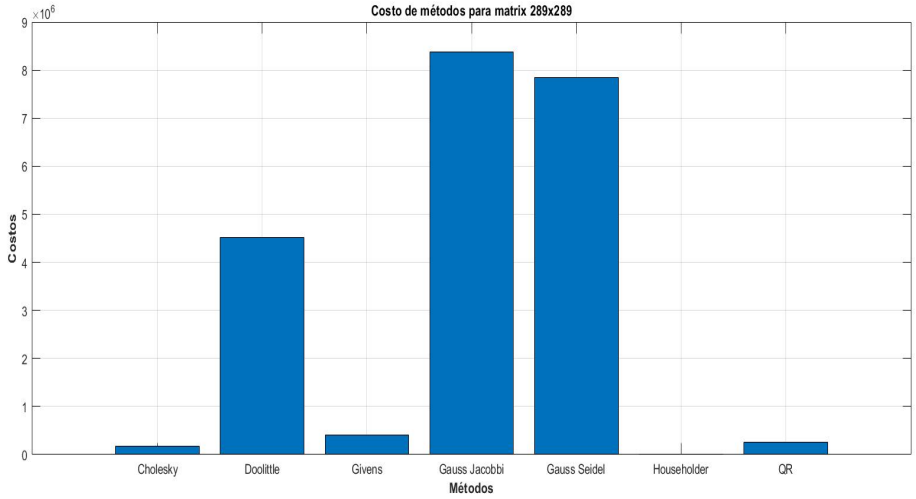

Figura 3-22: Costos operacionales en matríz 289x289

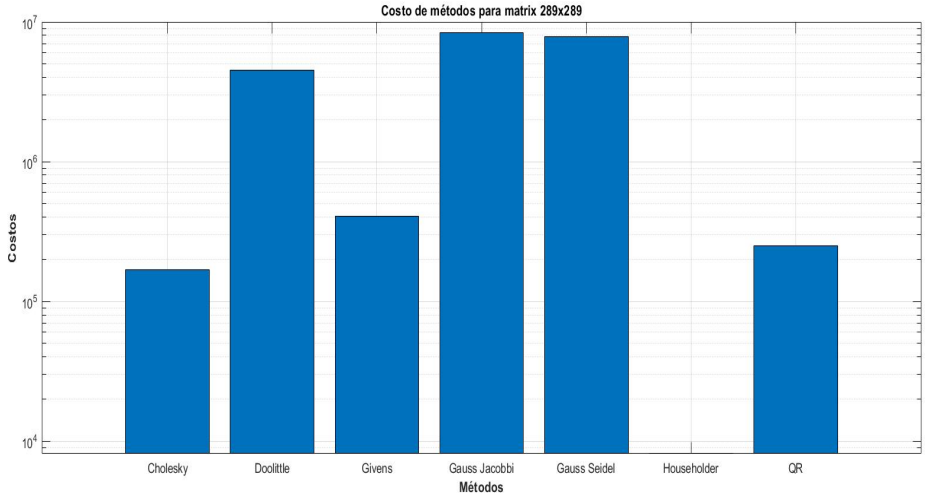

Figura 3-23: Costos operacionales en matríz 289x289 con Y log

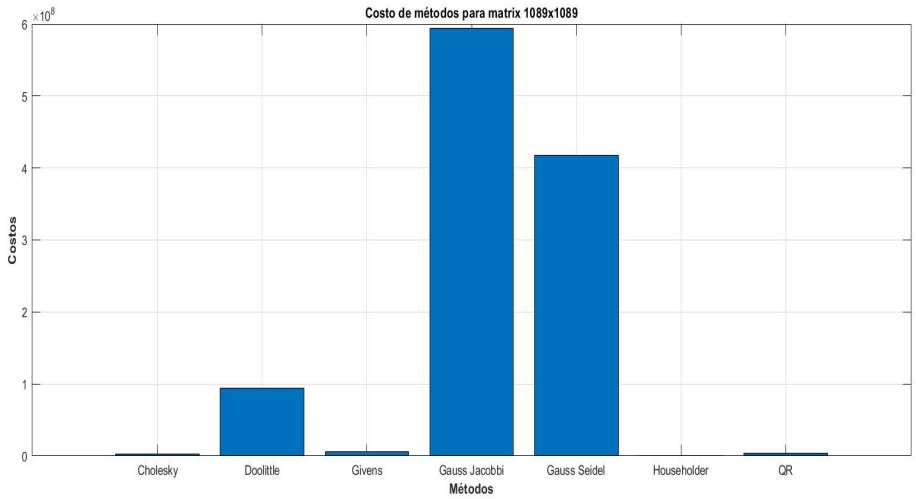

Figura 3-24: Costos operacionales en matríz 1089x1089

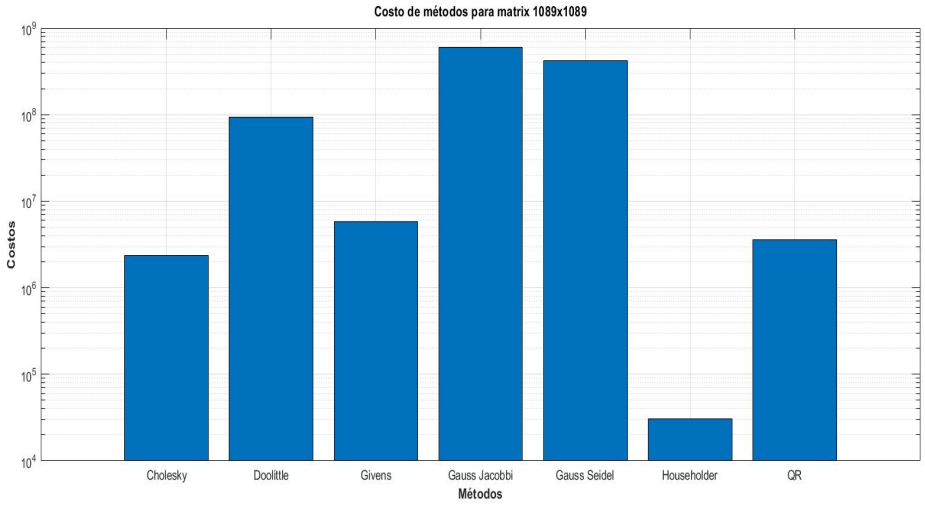

Figura 3-25: Costos operacionales en matríz 1089x1089 con Y log

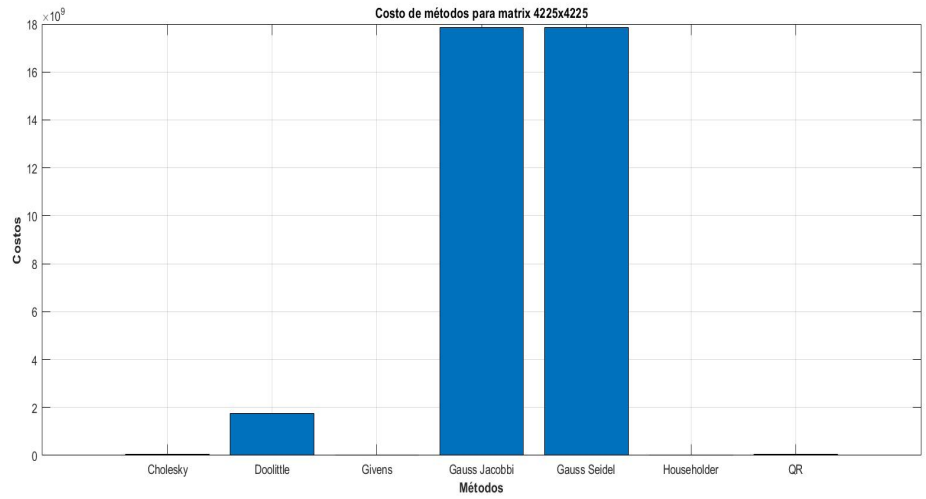

Figura 3-26: Costos operacionales en matríz 4225x4225

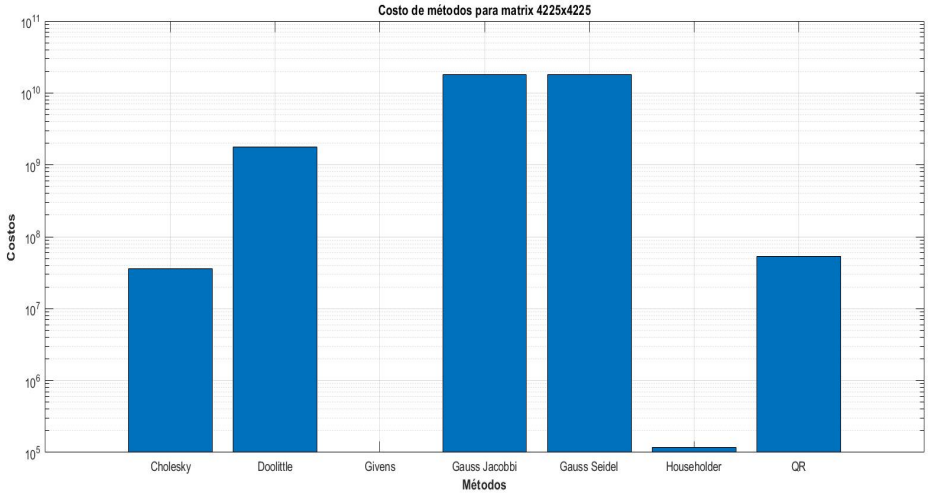

Figura 3-27: Costos operacionales en matríz 4225x4225 con Y log

3.4 GESTOR DE MÉTODOS VS EJECUCIÓN DE TODOS LOS MÉTODOS

Tabla 3.1: Tiempos de ejecución con gestor y sin gestor, medido en segundos.

Matriz	**Con gestor [s]**	**Sin gestor [s]**
289x289	0.265331800000000	19.998209500000000
1089x1089	10.78378870000000	17994,501750000000
4225x4225	181.9147914000000	2227,1316020000000

En la *Tabla 3.1* se aprecian los valores de tiempos de ejecución para cada una de las matrices haciendo uso del gestor y sin hacer uso de él.

En el caso de la matriz de 289x289, al hacer uso del gestor, el método seleccionado es Cholesky, el que tiene un error de 3.151068107105793.

Para la matriz de 1089x1089, el gestor nuevamente selecciona a Cholesky, con un error de 5.283192595769832.

Finalmente para la matriz de 4225x4225, el gestor selecciona Gauss - Seidel para la ejecución del código, con una error de 1.299843410368742e-06

CAPÍTULO 4. ANÁLISIS DE LOS RESULTADOS

4.1 NEWTON MULTIVARIABLE

En la *Figura 3-1*, se puede apreciar claramente cómo hace Newton Multivariable para encontrar las soluciones de las variables. Partiendo desde 0, los valores en un principio parecen dispararse, pero luego se ve cómo x_1, x_2 y x_3 van convergiendo al número que finalmente es la solución del sistema.

Es importante notar que en sólo 12 iteraciones, Newton Multivariable fue capaz de encontrar la solución al sistema con un error nulo. De hecho, en la *Figura 3-3* el gráfico muestra como el error decae de manera prácticamente exponencial, acercándose rápidamente al 0.

Esto puede dar una idea de cómo, dependiendo de qué tan bueno sea el punto inicial que se le entregue al método y el tipo de funciones que debe analizar, éste es capaz de encontrar una solución en cuestion de segundos. Ahora, se debe conciderar que las funciones no era demasiadas, lo que cláramente facilita el cálculo, ya que no es lo mismo tener 3 ecuaciones no lineas a tener un 1 millon de ellas. En este último caso, sería necesario una gran cantidad de iteraciones, o asignar una tolerancia de error estimada para obtener una solución que sea lo suficientemente eficaz.

Finalmente, en la *Figura 3-2* es posible ver cómo las funciones se van acercando a 0 al ser evaluadas en las variables x_1, x_2 y x_3, considerando que ya estando cercanos a la iteración 6 o 7, el error decae considerablemente.

4.2 GAUSS - SEIDEL VS GAUSS - JACOBI

En los diferentes experimentos realizados para verificar los errores de Gauss - Seidel y Gauss - Jacobi, dan por ganador a Gauss - Seidel como el método que converge más rápido, en la menor cantidad de iteraciones.

Ahora, es interesante analizar cómo a medida que la matriz crece en tamaño, las diferencias entre estos métodos van disminuyendo, pero, es notorio cómo Gauss - Jacobi se va volviendo asintótico al eje X en la *Figura 3-9*, por lo que con una gran cantidad de iteraciones, muy probablemente se quedará estancado en un error y no bajará de ahí, o si

lo hace, será ínfimo.

De esta manera, usando como criterio el error y la convergencia a este, se puede apreciar que Gauss - Seidel es mejor en cada uno de los casos, lo que lo deja como un fuerte candidato a utilizar para matrices de gran tamaño. Ahora, es importante mencionar que Gauss - Seidel cuenta con más restricciones que Gauss - Jacobi, ya que para converger nececita que se cumpla una de las siguientes propiedades en la matriz entrante:

1. La matriz es diagonal dominante.

2. La matriz es simétrica y definida positiva al mismo tiempo.

Como cada una de las matrices de prueba cumplían con lo anterior mencionado, Seidel fue la mejor opción dentro de los métodos iterativos, pero de no haber sido así, es probable que Jacobi hubiese sido el método que mejor se comporta.

4.3 COMPARACIÓN MÉTODOS ITERATIVOS Y MÉTODOS DIRECTOS

Para esta sección es necesario tomar en consideración varios aspectos, tales como el error de los métodos, su tiempo de ejecución y los costos operacionales que conllevan cada uno.

1. **Análisis errores (eficacia):**

   En las pruebas con la matriz de 289x289, se puede apreciar en la *Figura 3-11*, pero más claramente en la *Figura 3-10* cómo cholesky es el método que tiene un mayor error con respecto a sus pares, siendo Gauss - Jacobbi y Gauss - Seidel los métodos que lo siguen. Cláramente lo que ocurre con Cholesky no es un error normal, por lo que su gran valor en comparación a los otros métodos, puede deberse a un error en su progamación o en la manera en que se calcula el error. Es curioso ver que Givens y Householder tienen un error mayor a QR y Doolittle, teniendo en consideración que la teoría menciona que estos métodos son los que tienen una mayor estabilidad numérica. Este fenómeno se puede dar por varias razones, pero las más posibles, son las mismas que podrían estar afectando a Cholesky.

De esta manera, se puede decir que el método más eficaz para la matriz de 289x289 es Doolittle, mientras que el menos eficaz sería Cholesky. Gauss - Seidel y Gauss - Jacobi tienen una eficacia de orden de magnitud bastante similar, mientras que Givens, Householder y QR siguen bastante de cerca a Doolittle.

Un patrón similar se aprecia en las pruebas para las otras matrices, pero con un dato que llama la atención en la *3-13*. El último error calculado para Gauss - Seidel es de 0, es decir, no hay error, lo que indica una gran eficacia para esta matriz en comparación a cualquier otro método utilizado, mientras que Cholesky nuevamente figura como el método menos eficaz.

En el caso de la matriz de 4225x4225, es interesante ver como Gauss - Jacobi se acerca al método de Cholesky en la *Figura 3-15*, al menos, en orden de magnitud, lo que los cataloga como los métodos menos eficaces para este caso. Por otro lado, Doolittle nuevamente es identificado como el método más eficaz.

2. **Análisis de tiempos de ejecución y costos operacionales (eficiencia):**

   Para medir la eficiencia de los métodos se considerarán los tiempos de ejecución y los costos operacionales mostrados anteriormente en el apartado de Resultados.

   La primera medición se tiene para la matriz de 289x289, en la que tanto en la *Figura 3-16* como en la *Figura 3-17* se puede apreciar como Givens es por lejos la que más tiempo demora en ejecutarse, mientras que, los demás métodos tienen resultados similares, siendo Gauss - Seidel el que finaliza su tarea en primer lugar. Ahora, si consideramos los costos operacionales de la *Figura 3-22* y *Figura 3-23*, se puede apreciar cómo Gauss - Seidel y Gauss - Jacobi son los métodos que más operaciones realizan, mientras que Householder es la que menos costos operacionales conlleva.

   Haciendo uso de los tiempos de ejecución y los tiempos operacionales, no es fácil ver qué método es el más o menos eficiente, pero lo que sí es claro, es que Givens es de los menos eficientes debido a su largo tiempo de ejecución en comparación a sus otros métodos, por lo que su costo operacional puede ser engañoso, ya que lo que constantemente se está realizando dentro de este método, es una serie de

multiplicaciones de matrices que son muy costosas, mucho más de lo que lo sería una simple suma o resta, pero que para efectos prácticos de este laboratorio, se consideraron como iguales.

Dado que el método de Householder tiene un bajo tiempo de ejecución y una aún más baja cantidad de costos operacionales, se puede indicar que este método es el más eficiente para esta matriz.

Ahora, tomando en cuenta los resultados para la matriz de 1089x1089 de la *Figura 3-18* y *Figura 3-19* Householder pierde el podio de ser el método más eficiente, siendo superado cláramente por Doolittle, a pesar de que sus costos operacionales no sean los menores, como se puede apreciar en la *Figura 3-25*. Sin embargo, Givens vuelve a mostrarse como el método menos eficiente, superando con creces los tiempos de ejecución de los otros métodos.

Finalmente, para la matriz de 4225x4225 Givens fue un caso perdido. Luego de 3 días de espera a que el método acabara, se decidió quitarlo de este caso. Para los demás métodos, se puede apreciar en la *Figura 3-20* como Householder es quien tiene el mayor tiempo de ejecución, dando cuenta también en la *Figura 3-21*, que todos los otros métodos (exceptuando Givens, claramente), tuvieron un tiempo de ejecución similar, siendo Doolittle ligeramente menor al resto. Ahora, si se mira la *Figura 3-27*, se puede apreciar como contrariamente a lo que se creería, Householder es el método con menos costos operacionales, pero esto al igual como ocurre con Givens, puede ser engañoso, ya que para encontrar la solución de los sistemas, Givens hace uso de reflexiones, las que tienen un gran costo operacional. Es por esto que para esta matriz, luego de Givens por supuesto, se puede considerar como el método menos eficiente, mientras que Cholesky, debido a su bajo costo operacional en comparación a los demás métodos, puede ser considerado como el más eficiente para esta matriz.

4.4 ANÁLISIS GESTOR DE MÉTODOS

Se puede apreciar cláramente en la *Tabla 3.1* como el gestor es realmente una mejora cuantitativa en cuanto la eficiencia, ahora, como el método al que entra es Cholesky y existen ciertos problemas con respecto a su error, la eficacia del gestor no es la mejor, a excepción cuando la matriz es de 4225x4225 que se ejecuta Gauss - Seidel y el error es considerablemente menor.

CAPÍTULO 5. CONCLUSIONES

Si bien se pudieron implementar todos los métodos para este laboratorio, es claro que no todos fueron implementados de la mejor manera, ya que ciertos errores no tienen mucha explicación. Por otro lado, Givens fue un gran problema al momento de realizar las pruebas, debido a sus largos tiempos de ejecución.

Este laboratorio ha sido todo un desafío debido a la complejidad que puede llegar a tener y a pesar de que hubo una extensión del plazo de entrega, hubiese sido ideal tener un par de días más para afinar algunos detalles del informe, aún así, considero que ha sido una experiencia enriquecedora y que por primera vez da un acercamiento real de la ingeniería en términos de los números y cómo se trabaja con ellos en la industria. Ya no sirve realizar las cosas de manera análoga, ahora todo debe ser programado para procesar miles y millones de datos, pero no sólo procesarlos, si no, también darles una interpretación que permitan al ingeniero tomar decisiones u observar fenómenos que ocurren en la operatoria de los métodos.

CAPÍTULO 6. BIBLIOGRAFÍA

Chapra, S. & Canale, R. (2007). *Métodos Numéricos para Ingeniería* (5th ed.). Mc. Graw Hill.

Cobos, F. (2004). *Apuntes de Álgebra Numérica*. Universidad de Sevilla.