

INFORME DE LABORATORIO 1 DE ALGORITMOS NUMÉRICOS:
MÉTODOS NUMÉRICOS

ISAAC ESPINOZA

Profesor: Óscar Rojas Díaz.

Santiago - Chile
13 de mayo de 2019

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS.....	v
ÍNDICE DE CUADROS	vi
CAPÍTULO 1. INTRODUCCIÓN	7
1.0.1 Objetivos	7
1.0.2 Estructura del Informe	8
CAPÍTULO 2. MARCO TEÓRICO	9
2.1 MÉTODOS DE ECUACIONES NO LINEALES	9
2.1.1 Newton Raphson Multivariable	9
2.2 MÉTODOS DE SISTEMAS DE ECUACIONES LINEALES	9
2.2.1 Métodos Directos	10
2.2.2 Método LU y Cholesky	11
2.2.3 Método QR, Givens y Householder	11
2.2.4 Métodos iterativos	12
2.2.5 Gauss-seidel y Gauss-jacobi	12
CAPÍTULO 3. FUNCIONES Y SISTEMAS DE ECUACIONES.....	15
CAPÍTULO 4. ANÁLISIS DE RESULTADOS.....	19
4.1 ANÁLISIS NEWTON RAPHSON	19
4.2 ANÁLISIS MÉTODOS ITERATIVOS Y DIRECTOS	21

4.2.1	Análisis costo operacional y complejidad para métodos iterativos y directos	21
4.2.2	Análisis resultados y errores métodos directos e iterativos	22
4.3	ANÁLISIS DE APLICACIÓN DEL ALGORITMO DE DECISIÓN . . .	36
CAPÍTULO 5. CONCLUSIONES		43
	Bibliografía	46

ÍNDICE DE FIGURAS

4.1	Gráficos aproximación y errores método Newton Raphson.	20
4.2	Gráficos tiempos, matriz 289x289.	23
4.3	Gráficos tiempos, matriz 1089x1089.	24
4.4	Gráficos tiempos, matriz 4225x4225.	25
4.5	Gráficos resultados, matriz 289x289.	27
4.6	Gráficos resultados, matriz 1089x1089.	27
4.7	Gráficos resultados, matriz 4225x4225.	28
4.8	Gráficos error, matriz 289x289.	28
4.9	Gráficos error, matriz 1089x1089.	29
4.10	Gráficos error, matriz 4225x4225.	29
4.11	Gráficos resultados métodos directos, matriz 289x289.	30
4.12	Gráficos resultados métodos directos, matriz 1089x1089.	31
4.13	Gráficos resultados métodos directos, matriz 4225x4225.	32
4.14	Gráficos aproximación 2D métodos iterativos, matriz 289x289.	33
4.15	Gráficos aproximación 2D métodos iterativos, matriz 1089x1089.	34
4.16	Gráficos aproximación 2D métodos iterativos, matriz 4225x4225.	35
4.17	Tiempos de aplicación del algoritmo para todas las matrices.	39
4.18	Tiempos de verificación de propiedades de matrices.	40

ÍNDICE DE CUADROS

Tabla 4.1: Resultados método Newton Raphson Multivariable	19
Tabla 4.2: Tabla costo operacional según método y matriz	22
Tabla 4.3: Número de Iteraciones y Error, Métodos Iterativos	26
Tabla 4.4: Errores métodos directos vs métodos iterativos	36
Tabla 4.5: Matrices y propiedades	37
Tabla 4.6: Matrices y métodos	37
Tabla 4.7: Tabla costo operacional según método y matriz, algoritmo	38

CAPÍTULO 1. INTRODUCCIÓN

La experiencia de laboratorio número 1 consiste en la implementación de una serie de métodos numéricos para ecuaciones no lineales, así como también para sistemas de ecuaciones lineales, específicamente métodos directos e iterativos, como por ejemplo Gauss Jacobi, Gauss Seidel, Hoshoulder, entre otros. Además de la implementación, se debe implementar un algoritmo que determine ciertas características de una matriz (sistema de ecuación lineal) y en base a dichas características escoja el mejor método directo o iterativo dentro de los disponibles. Finalmente, se deben analizar los resultados y presentar gráficos correspondientes en caso de ser necesario.

Al intentar aplicar las matemáticas a problemas del mundo real, muchas veces pueden surgir problemas que no pueden ser resueltos de forma exacta y por lo tanto la resolución de estos problemas debe ser abordada con métodos numéricos. Además, se debe reconocer que no todos los problemas pueden ser abordados por los mismos métodos numéricos, por lo tanto es importante conocer las características y propiedades de los problemas que permiten que sean resueltos con un método específico o en el caso de tener múltiples opciones, saber reconocer cual sería mas eficiente o eficaz. A grandes rasgos, en esta experiencia se abordarán problemas de ecuaciones no lineales y multivariantes a través del método Newton Raphson multivariable. También se resolverán problemas de sistemas de ecuaciones lineales, estos problemas tienen la dificultad de que se componen de muchas variables y se debe decidir si es mejor abordarlos con un método directo o con un método iterativo (analizando sus pros y contras).

1.0.1 Objetivos

Se tienen los siguientes objetivos:

- Implementar método de Newton Raphson multivariable
- Implementar los métodos iterativos Gauss Seidel y Gauss Jacobi
- Implementar los métodos directos QR, Householder, Givens, LU y Cholesky

- Implementar un algoritmo que decida qué método es mejor utilizar dada una matriz (solo para sistemas de ecuaciones lineales)
- Analizar los resultados obtenidos a través de gráficos y métricas de tiempo y complejidad algorítmica

1.0.2 Estructura del Informe

El informe se compone de las siguientes secciones:

1. Marco teórico con descripción de los métodos que se utilizaron
2. Descripción de los datos disponibles (función y matrices)
3. Análisis de resultados, eficiencia y complejidad
4. Conclusión

CAPÍTULO 2. MARCO TEÓRICO

A continuación se describirán brevemente los métodos de ecuaciones no lineales y métodos directos e iterativos para funciones lineales.

2.1 MÉTODOS DE ECUACIONES NO LINEALES

Uno de los problemas básicos es la búsqueda de raíces exactas o aproximadas en ecuaciones no lineales, si bien existen muchos métodos para esto, para esta experiencia solo se utiliza el método de Newton Raphson para múltiples variables.

2.1.1 Newton Raphson Multivariable

Este método se basa en el método de Newton para encontrar aproximaciones de las raíces de una o más funciones de una o más variables, así como también se pueden encontrar máximos y mínimos de funciones encontrando sus ceros en sus derivadas respectivas. Se aplica sobre sistemas de ecuaciones de forma:

$$f_1(x_1, x_2, \dots, x_n) = 0, f_2(x_1, x_2, \dots, x_n) = 0 \dots f_n(x_1, x_2, \dots, x_n) = 0 \quad (2.1)$$

Se necesita obtener el jacobiano inverso del sistema de ecuaciones y en un proceso iterativo se puede obtener una aproximación de las raíces si se cumple con el criterio de convergencia.

El criterio de convergencia supone lo siguiente:

$f_1(r_1, r_2, \dots, r_n) = f_2(r_1, r_2, \dots, r_n) = \dots = f_n(r_1, r_2, \dots, r_n) = 0$, además con el determinante del jacobiano del sistema de ecuaciones distinto de 0 y las derivadas parciales segundas son continuas, se tiene un conjunto en el cual el método de Newton Raphson Multivariable converge al punto (r_1, r_2, \dots, r_n) .

Cabe destacar que debido al carácter iterativo de este método, se suelen usar condiciones de paradas como tolerancia o número de iteraciones máxima, por lo tanto es posible no obtener una solución exacta sino que una aproximación que tiene un error asociado.

2.2 MÉTODOS DE SISTEMAS DE ECUACIONES LINEALES

Los métodos para sistemas de ecuaciones lineales, que pueden ser directos o iterativos según las características del sistema, buscan raíces o aproximaciones para sistemas que tienen la siguiente forma:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

Este sistema se puede expresar en forma matricial como:

$$Ax = b \quad (2.2)$$

Donde A es la matriz de coeficientes, x es la matriz transpuesta de las variables y b es la matriz transpuesta de los valores b_1, b_2, \dots, b_n . Debido a que tanto los métodos directos como los iterativos utilizan la representación matricial, estos también dependen de ciertas características de las matrices para poder ser aplicados sobre un sistema, estas características son:

1. Definida positiva
2. Definida semi-positiva
3. Negativa
4. Simetría
5. Diagonal dominante
6. Dispersión

2.2.1 Métodos Directos

Los métodos directos básicamente son métodos algebraicos para resolver sistemas de ecuaciones lineales que buscan reducir el sistema original a uno más simple de resolver. Estos métodos se basan en lo siguiente:

Sean A, B dos matrices cuadradas que representan sistemas de ecuaciones lineales,

$Ax = b$ y $Bx = d$ son equivalentes si poseen exactamente las mismas soluciones.

Por lo tanto, si se tiene un sistema de la forma $Ax = b$, se puede reducir a su forma más simple $Bx = d$ y por lo tanto se obtienen sus soluciones.

2.2.2 Método LU y Cholesky

Se tiene un sistema de ecuaciones lineales de forma $Ax = b$. El método de factorización LU consiste en la factorización de la matriz A en dos matrices. La matriz L es una matriz triangular inferior, mientras que la matriz U es una matriz triangular superior. Por lo tanto, del sistema original, se pasa a lo siguiente: $LUx = b$, con $Ux = z$ y $Lz = b$, donde primero se resuelve el sistema $Lz = b$ y luego $Ux = z$, obteniendo finalmente la solución al sistema original. Cabe destacar que el método LU se puede abordar con el método Doolittle o el método Crout, siendo la única diferencia entre ambos que el método Doolittle considera la diagonal de la matriz L igual a 1, mientras que el método Crout considera la diagonal de la matriz U igual a 1.

Respecto al método Cholesky, es un caso particular en el cual si la matriz A es simétrica y definida positiva, se puede descomponer como $A = LL^*$, siendo L la matriz triangular inferior (igual al concepto de L en el método LU) y L^* la conjugada transpuesta de L . Cabe destacar que el método LU se puede aplicar en matrices cuadradas que tengan una inversa, es decir, su determinante sea distinto de 0, mientras que el método Cholesky se debe aplicar en matrices definidas positivas y simétricas.

2.2.3 Método QR, Givens y Householder

El método QR tiene la particularidad de poder usarse sobre matrices que no son rectangulares además de matrices cuadradas, a diferencia de todos los métodos vistos anteriormente. Consiste en una factorización de la forma $A = QR$, con Q una matriz con columnas ortonormales y R una matriz triangular superior. ([4])

Este método se suele utilizar para obtener vectores y valores propios de una matriz, pero también se puede utilizar como un método de resolución directo de sistemas de ecuaciones lineales. De la misma forma que los métodos directos LU, se tiene un sistema de ecuaciones que se puede expresar en la forma matricial $Ax = b$, por lo tanto para resolverlo se necesita aplicar una factorización QR y finalmente resolver el sistema resultante $Rx = QTb$. Se puede considerar como desventaja que este tipo de factorización es más costoso que la factorización LU. La descomposición QR se suele realizar con el método de ortogonalización de Gram-Schmidt, pero también se puede implementar la factorización

QR utilizando reflexiones de Householder o rotaciones de Givens.

Las diferencias entre estos métodos más que nada se relacionan con la inestabilidad numérica (producida por ejemplo al utilizar Gram-Schmidt), la facilidad para implementar el algoritmo de factorización y el gasto operacional de este.

Factorizando con Gram-Schmidt, se tiene: "Consiste en encontrar una base ortonormal de las columnas de A. La columna $k + 1$ de Q se genera como una combinación lineal de la columna $k + 1$ de A y las primeras k columnas ya construidas de Q, tal que los coeficientes de la combinación lineal favorezcan que dicho vector sea ortogonal a los anteriormente construidos."([1])

Cabe destacar que Gram-Schmidt es la forma más numéricamente inestable (entre Gram-Schmidt, Householder y Givens) de realizar una factorización QR. Las reflexiones de Householder son la forma más simple y numéricamente estable para realizar una factorización de forma QR, pero a nivel algorítmico es costoso y no se puede paralelizar. Finalmente, el método de las rotaciones de Givens es menos costoso que Householder y si puede ser paralelizable.

2.2.4 Métodos iterativos

Los métodos iterativos se utilizan para encontrar aproximaciones a sistemas de ecuaciones lineales, porque algunas veces (según las características o tamaño de una matriz) puede ser demasiado costoso buscar la solución exacta. Al ser métodos iterativos, su funcionamiento tiene que ver con la convergencia del método hacia la solución, que se puede lograr si es que la matriz que representa al sistema de ecuaciones posee características particulares. Estos métodos deben partir con una aproximación inicial que suele ser de la forma $x^{(0)}$ con valores arbitrarios o iguales a 0. A partir de dicho vector inicial, genera una sucesión de vectores x_1, x_2, \dots, x_n , siendo el método consistente con el sistema $Ax = b$ si se llega a una solución exacta, o convergente si la sucesión de vectores converge a la solución. ([5])

Si un método es consistente, entonces es convergente, pero si converge no necesariamente es consistente.

2.2.5 Gauss-seidel y Gauss-jacobi

- Gauss-jacobi: El método de Gauss-jacobi consiste en la descomposición de la matriz A , que se obtiene de $Ax = b$, en $A = M - N = D - (E + F)$, resultando en el sistema: $Ax = b \Rightarrow Dx = (E + F)x + b \Rightarrow x = D^{-1}(E + F)x + D^{-1}b$ ([5]). Finalmente, la matriz $J = D^{-1}(E + F)$ se denomina matriz de Jacobi, la cual también se expresa como $I - D^{-1}A$. Para que este método pueda converger, se necesita que la matriz A sea cuadrada y de diagonal dominante.

- Gauss-seidel: El método de Gauss-seidel es similar al método jacobi, para converger necesita la misma condición (matriz A diagonal dominante). La diferencia radica en que la descomposición del sistema $Ax = b$ se realiza de forma distinta, resultando: $A = M - N = (D - E) - F$. Despejando: $Ax = b \Rightarrow (D - E)x = Fx + b \Rightarrow x = (D - E)^{-1}Fx + (D - E)^{-1}b$, donde finalmente se tiene $L1 = (D - E)^{-1}F$, que se puede expresar como: $L1 = I - (D - E)^{-1}A$. ([5]).

CAPÍTULO 3. FUNCIONES Y SISTEMAS DE ECUACIONES

En esta experiencia se debe resolver un sistema de ecuación no lineal y tres sistemas de ecuaciones lineales, entregados en forma de matrices cuadradas.

$$f(x_1, x_2, x_3) : x_1^2 + x_2 - 37 = 0$$

$$g(x_1, x_2, x_3) : x_1 - x_2^2 - 5 = 0$$

$$h(x_1, x_2, x_3) : x_1 + x_2 + x_3 - 3 = 0$$

$$\text{Con } X_0 = (0, 0, 0)^T$$

Por lo tanto, se tiene un sistema de ecuación no lineal con valores iniciales iguales a $(0, 0, 0)$ que debe ser resuelto mediante el método Newton Raphson Multivariable, además en el enunciado no se pide un número específico de iteraciones, por lo tanto se debe intentar obtener el error mínimo.

Cabe destacar que utilizando un sistema operativo de 64 bits, el error máximo posible obtenible por Matlab corresponde a 2^{63} , debido a que 64 es la mayor cantidad de bits que Matlab les asigna a los datos de tipo punto flotante (en formato double), pero un bit es asignado para el signo del valor. Por lo tanto, 2^{63} es equivalente a 10^{18} y se considerará como tolerancia o menor error posible al valor $e = 10^{-18}$.

Se tienen tres sistemas de ecuaciones lineales entregados en forma de matrices, los cuales tienen las siguientes características generales:

1. Matriz cuadrada de 289x289
2. Matriz cuadrada de 1089x1089
3. Matriz cuadrada de 4225x4225

Además, se considera que las matrices (para efectos del laboratorio) pueden tener 10 propiedades, las cuales son las siguientes:

1. Matriz cuadrada: Una matriz es cuadrada si posee dimensiones $n \times n$, es decir su número de filas es igual a su número de columnas.
2. Definida positiva: Se dice que una matriz A es definida positiva si $Y^T A Y > 0$ para todo vector $Y \neq 0$
3. Definida semi positiva: Una matriz A se dice que es semidefinida positiva si $Y^T A Y \geq 0$ para todo vector $Y \neq 0$.
4. Diagonal dominante: Una matriz A es diagonal dominante por filas cuando, para todas las filas, el valor absoluto del elemento de la diagonal de esa fila es estrictamente mayor que la norma del resto de elementos de esa fila. Lo es por columnas cuando, para todas las columnas, el valor absoluto del elemento de la diagonal de esa columna es estrictamente mayor que la norma del resto de elementos de esa columna.
[11]
5. Definida negativa: Se dice que una matriz A es definida negativa si $Y^T A Y < 0$ para todo vector $Y \neq 0$.
6. Simétrica: Una matriz A es simétrica si $A = A^T$, es decir, si la matriz original es igual a su transpuesta.
7. Dispersa: Se considera como una matriz dispersa a una matriz A , cuya sumatoria de elementos $a_{ij} = 0$ sea mayor a la sumatoria de sus elementos $a_{ij} \neq 0$, es decir, que posee mas elementos iguales a 0 que elementos distintos a 0.
8. Pequeña: Se dice que un sistema de ecuaciones en su forma matricial es grande si su n (número de ecuaciones, equivalente a número de filas de la matriz resultante) es menor a 300. [5]
9. Grande: Se dice que un sistema de ecuaciones en su forma matricial es grande si su n (número de ecuaciones, equivalente a número de filas de la matriz resultante) es mayor a 300. [5]

10. Matriz rectangular: Una matriz es rectangular si posee dimensiones $m \times n$ con $n \neq m$, siendo m su número de filas y n su número de columnas.

Por teoría, se tiene que los métodos iterativos y directos que se utilizarán solo pueden aplicarse si se cumplen ciertas condiciones en la matriz objetivo, lo cual se especifica a continuación:

- Gauss-seidel: Como este método es iterativo, solo se busca llegar a una buena aproximación de los valores reales de la solución, por lo tanto se tienen condiciones de convergencia que están relacionadas con propiedades del sistema de ecuaciones. En el caso de gauss-seidel, se necesita que la matriz sea cuadrada y diagonal dominante para que el método converga [9].

Además se tienen condiciones no estrictas para un mejor funcionamiento de este método en comparación con métodos directos, estas condiciones son: matriz grande y dispersa (de todas formas se aplicará para la matriz pequeña si es que esta es diagonal dominante y dispersa).

- Gauss-jacobi: Se debe cumplir una condición de convergencia de la misma forma que en gauss-seidel, en este caso, se necesita que la matriz A sea diagonal dominante y cuadrada (condición estricta) [9].

Además, este método se ve favorecido para resolver matrices grandes (más de 300 ecuaciones) y dispersas (de todas formas se aplicará para la matriz pequeña si es que esta es diagonal dominante y dispersa).

- LU (Doolittle): El método de factorización solo requiere que la matriz objetivo sea cuadrada de dimensiones $n \times n$ y que además los n menores principales de la matriz A sean distintos de 0. Además, cabe destacar que para la factorización LU Doolittle se fija la diagonal de la matriz L en valores iguales a 1. Finalmente, como este es un método directo, se considera apto para matrices pequeñas.

- Cholesky: Si A es una matriz cuadrada con dimensiones $n \times n$, además si es simétrica y definida positiva, entonces posee una descomposición de Cholesky ([9]). Sólo en

el caso mencionado es conveniente resolver una matriz con el método de factorización de Cholesky, además para efectos del algoritmo de decisión, se considera que Cholesky al ser método directo conviene aplicarlo sobre matrices pequeñas.

- QR: La única característica relevante de este método, es que es aplicable tanto a matrices cuadradas como rectangulares. No posee otro tipo de restricciones. Se determina que es más conveniente utilizar QR para matrices pequeñas, debido a que es un método directo.

- Givens: Debido a su funcionamiento, las rotaciones de Givens suelen utilizarse en matrices dispersas. ([3])

Además, debido a que es un método directo, se determina que es más conveniente utilizar las rotaciones de givens para matrices pequeñas. Finalmente este método se puede usar tanto para matrices cuadradas como rectangulares.

- Householder: Puede ser usado tanto en matrices cuadradas como rectangulares. Además, en promedio tiene mejores resultados para matrices no dispersas que el método givens, pero esto no significa que no pueda ser utilizado para la resolución de matrices dispersas ([6]).

Por lo tanto, para el algoritmo de decisión se determina que Householder puede ser aplicado a matrices pequeñas (porque es método directo) densas y dispersas.

En base a dichas características y a un análisis aplicado sobre las matrices, se decidirá con un algoritmo que métodos pueden ocuparse y cuales son los mejores métodos para resolverlas y si es que es necesario aplicar todos los métodos programados o solo algunos. Cabe destacar que de todas formas se aplicarán todos los métodos sobre todas las matrices, pero el análisis de resultado y tiempo/costo operacional se hará de forma separada y finalmente se compararán los resultados.

CAPÍTULO 4. ANÁLISIS DE RESULTADOS

En esta sección, se analizan los resultados de la aplicación de todos los métodos para las matrices de 289x289, 1089x1089 y 4225x4225 sin ningún tipo de restricción. Además también se presentaran los resultados del uso del método de Newton Raphson para la resolución de un sistema de ecuaciones no lineal.

4.1 ANÁLISIS NEWTON RAPHSON

El sistema de ecuaciones no lineales

$$f(x_1, x_2, x_3) : x_1^2 + x_2 - 37 = 0$$

$$g(x_1, x_2, x_3) : x_1 - x_2^2 - 5 = 0$$

$$h(x_1, x_2, x_3) : x_1 + x_2 + x_3 - 3 = 0$$

$$\text{Con } X_0 = (0, 0, 0)^T$$

se resolvió aplicando el método de Newton Raphson Multivariable. Como ya se mencionó, se utilizó una tolerancia de 10^{-18} debido a que es el valor mínimo que puede tomar un float double en matlab y arquitecturas de 64 bits. Además, se utilizó un tope de 500 operaciones en caso de que el método no pudiera converger a la tolerancia deseada.

Los resultados obtenidos fueron los siguientes:

Tabla 4.1: Resultados método Newton Raphson Multivariable

	Iteraciones										
Variables	1	2	3	4	5	6	7	8	9	10	11
x1	5	4,350	5,363	5,696	5,882	5,966	5,995	5,999	5,999	5,999	6
x2	37	18,491	9,255	4,665	2,427	1,412	1,058	1,001	1,000	1,000	1
x3	-39	-19,842	-11,619	-7,361	-5,310	-4,378	-4,053	-4,001	-4,000	-4,000	-4

Cabe destacar que los datos fueron truncados para poder visualizarse en la tabla, por lo tanto en las repeticiones de datos antes de la iteración número 11, en realidad si hay diferencias mínimas. Por lo tanto, como se puede ver en la tabla, el valor de la variable x1 fue aproximado a 6, el valor de la variable x2 a 1 y el valor de la variable x3 fue aproximado a -4, todo esto se realizó en 11 iteraciones. Recordando, el límite de iteraciones era de 500, para evitar una ejecución infinita del programa en caso de que no convergiera, por lo tanto

se puede determinar que se alcanzó la tolerancia de 10^{-18} y que los resultados obtenidos son exactos.

A continuación, se muestra gráficamente el comportamiento de la aproximación y del error obtenido con el método Newton Raphson: Analizando los resultados de los gráficos en

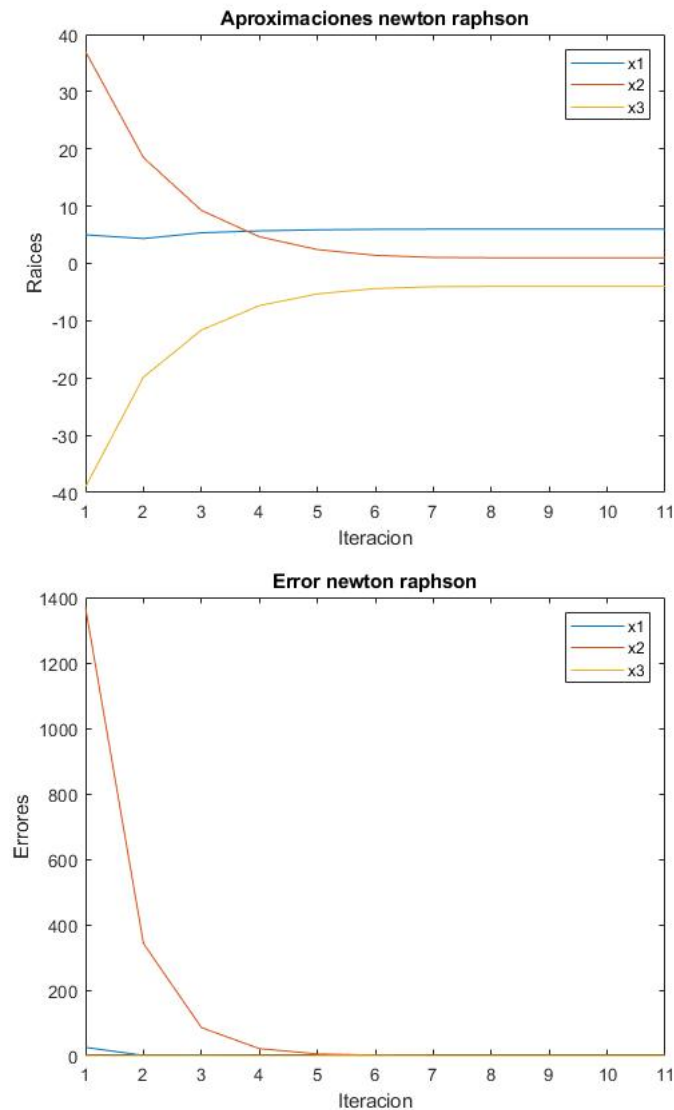


Figura 4.1: Gráficos aproximación y errores método Newton Raphson.

conjunto con la tabla anterior, se confirma que el error y las aproximaciones prácticamente convergieron a valores cercanos a los exactos en la iteración número 6, por lo tanto las

siguientes iteraciones solo se realizaron debido a la tolerancia, que buscaba el encontrar un error mínimo. En este caso, el error es mínimo y es igual a 0.

4.2 ANÁLISIS MÉTODOS ITERATIVOS Y DIRECTOS

4.2.1 Análisis costo operacional y complejidad para métodos iterativos y directos

En la siguiente tabla se estudia el costo operacional obtenido por cada método calculado con un contador correspondiente en cada función, además, por teoría se sabe lo siguiente:

- LU (Doolittle): Se tiene un costo operacional de $\frac{2}{3} * n^3$ operaciones de punto flotante, siendo n el tamaño de la matriz. ([10])
Además, se tiene que la complejidad de este algoritmo es de $O(n^3)$.
- Cholesky: Cholesky tiene una complejidad del orden $O(\frac{n^3}{3})$ ([2])
- QR: Este método resuelto sin ocupar rotaciones de givens o transformaciones de householder, tiene complejidad entre $O(n^3)$ y $O(n^4)$. ([8])
- Householder: Se tiene un costo operacional de $\frac{4}{3} * n^3$ operaciones de punto flotante, siendo n el tamaño de la matriz (considerando una matriz cuadrada). ([10])
Además, se tiene que la complejidad de este algoritmo es de $O(n^3)$.
- Givens: Tiene una complejidad de $O(n^3)$, con un costo operacional aproximado de $3mn^3 - n^3$, siendo n y m las dimensiones de la matriz en que se aplique las rotaciones de givens.
- Gauss Seidel y Gauss Jacobi: Su complejidad depende tanto de las dimensiones de la matriz como la cantidad de iteraciones realizadas para llegar a la tolerancia esperada. Se requiere cerca de $O(n^2)$ operaciones por iteración.

Luego, al analizar y comparar los resultados con la teoría, se tienen resultados satisfactorios:

Tabla 4.2: Tabla costo operacional según método y matriz

Métodos	289x289	1089x1089	4225x4225
LU Doolittle	16217813	862760613	50306048325
Cholesky	8170608	432266208	2.51e+10
QR	24429752	1.29e+09	1.78e+10
Householder	32394593	1.72e+09	1.006e+11
Givens	3508311	49809911	749730615
Gauss Jacobi	14918827	800047179	1.78e+10
Gauss Seidel	7710706	434446394	1.78e+10

Todos los resultados obtenidos de costo operacional se corresponde con las complejidades de cada algoritmo, excepto en el caso de Givens, en el que se obtienen menos operaciones de las esperadas. Cabe destacar que los resultados de Jacobi y Seidel se ven afectados por el número de iteraciones que cada uno realizó, lo cual depende de la tolerancia utilizada de 10^{-18} . Respecto a los tiempos de ejecución, se obtuvieron los siguientes resultados:

Como se puede observar en las figuras 4.4, 4.2 y 4.3, para el caso de los métodos directos se puede observar un comportamiento constante en las matrices de 289×289 y 1089×1089 . Para la matriz de 289×289 todos los métodos directos e iterativos presentaron tiempos triviales menores a los 2 segundos, siendo los más ineficientes Householder y Gauss Jacobi. Para la matriz de 1089×1089 , nuevamente Householder y Jacobi son los métodos más ineficientes, pero esta vez la diferencia entre el peor método directo y el peor método iterativo aumentó mucho. Finalmente, para la matriz de 4225×4225 , en el caso de los métodos directos se mantuvo Householder como el peor, pero QR empeoró su rendimiento, mientras que Givens lo mejoró respecto a doolittle, cholesky y QR. En el caso de los métodos iterativos, Seidel se demoró más tiempo que Jacobi a diferencia de las dos matrices anteriores.

Respecto a las grandes diferencias entre los métodos directos e iterativos, estas se pueden atribuir a que el número de iteraciones utilizado fue muy grande o la tolerancia muy pequeña.

4.2.2 Análisis resultados y errores métodos directos e iterativos

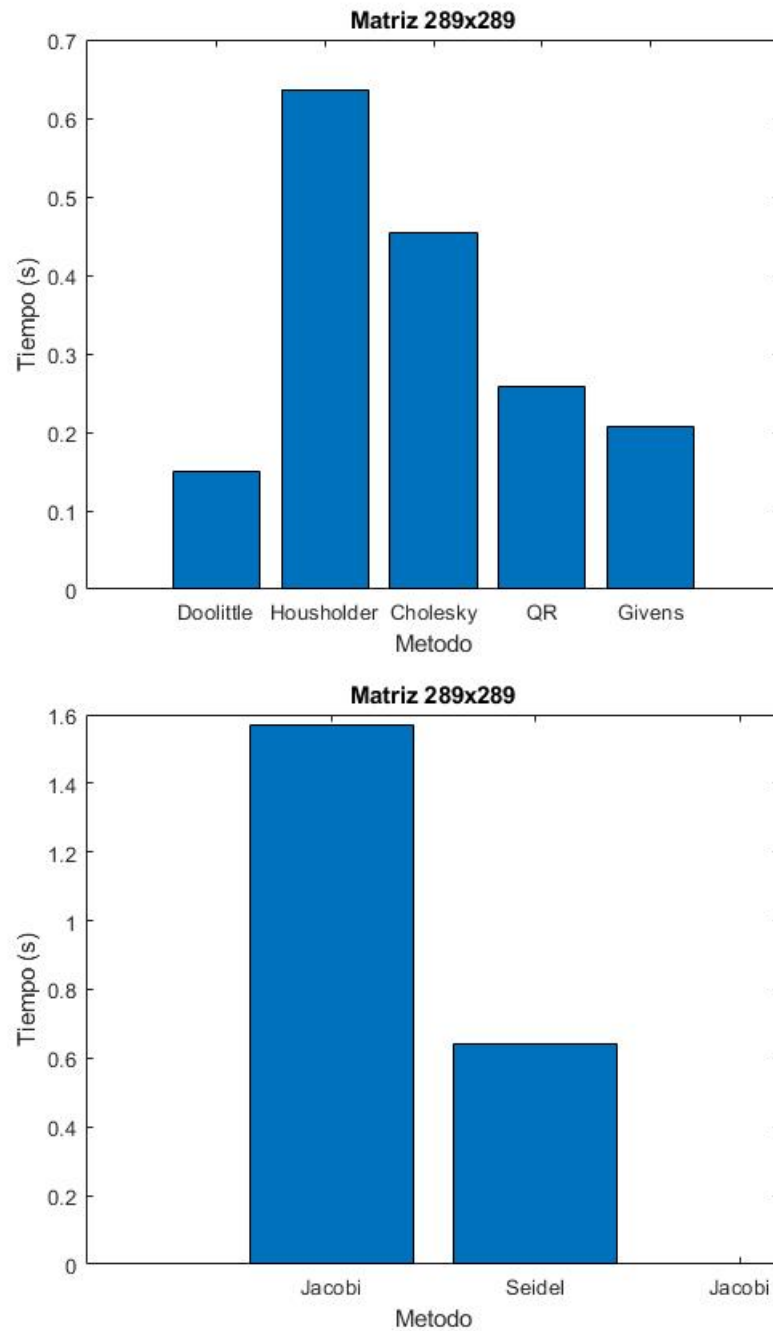


Figura 4.2: Gráficos tiempos, matriz 289x289.

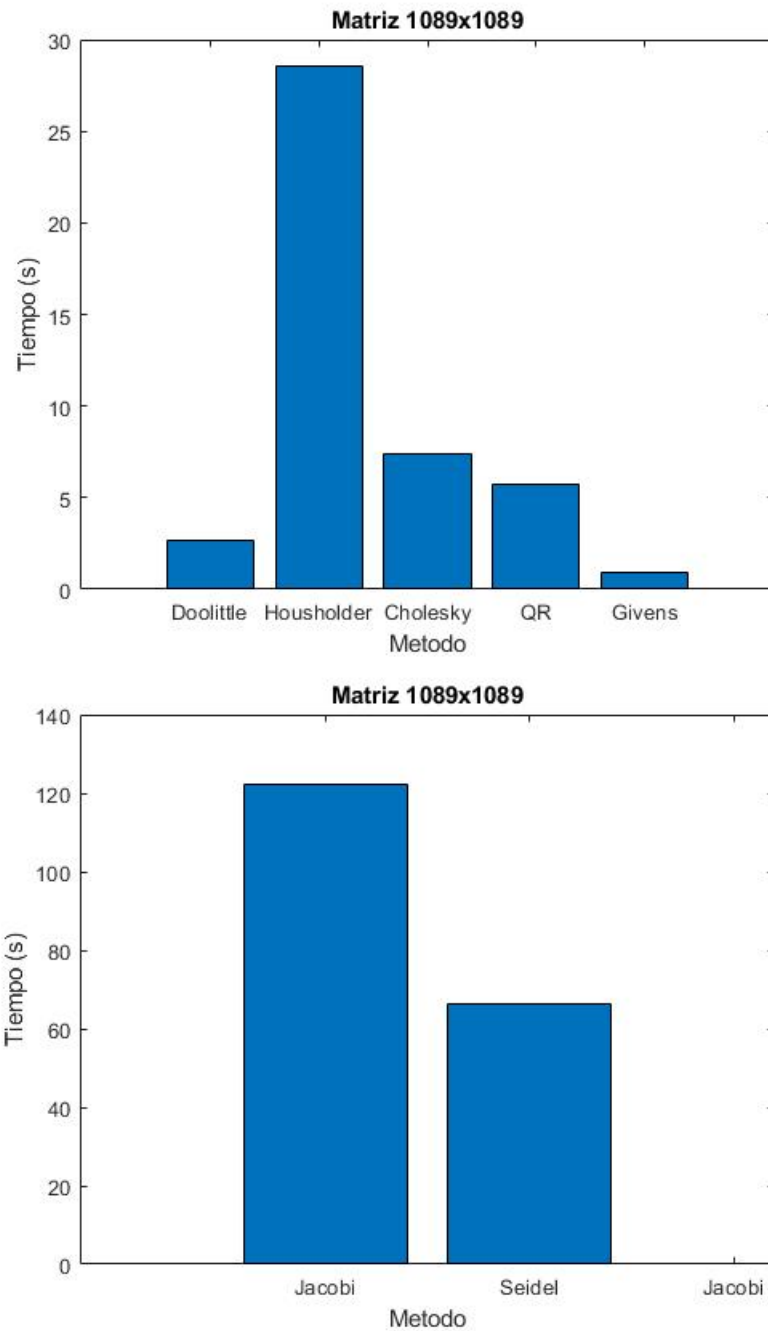


Figura 4.3: Gráficos tiempos, matriz 1089x1089.

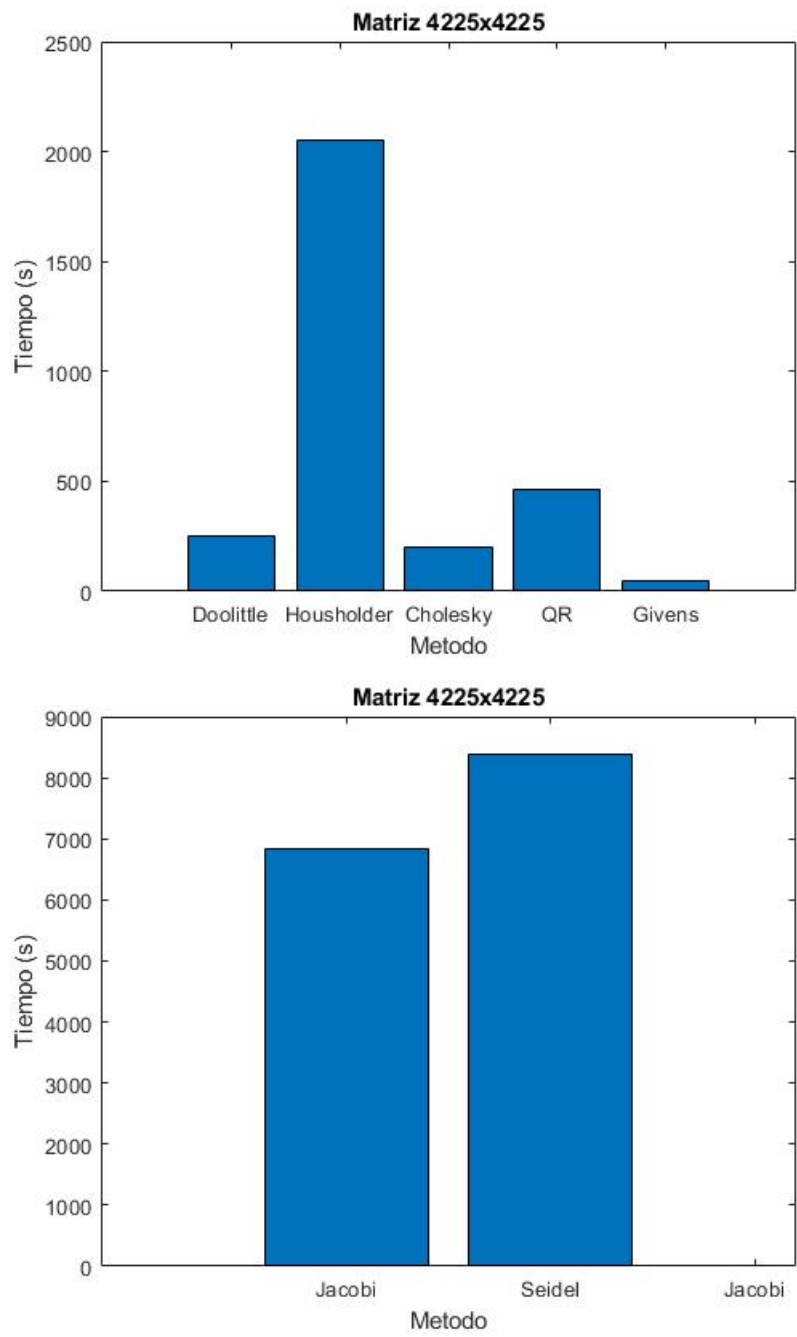


Figura 4.4: Gráficos tiempos, matriz 4225x4225.

Para analizar los resultados obtenidos, se hicieron gráficos correspondientes a los métodos directos e iterativos, debido a que en el caso de los métodos directos solo se tiene un set de resultados, mientras que los métodos iterativos tienen una aproximación de la solución que se puede representar como una matriz, en la cual cada columna nueva es la siguiente aproximación, siendo la columna final la aproximación final correspondiente a la iteración final y por lo tanto el resultado entregado. Como ya se mencionó, se utilizó una tolerancia de 10^{-18} y además un límite de 1000 iteraciones.

A continuación, se muestra una tabla de iteraciones de los métodos iterativos:

Tabla 4.3: Número de Iteraciones y Error, Métodos Iterativos

Métodos	289x289	1089x1089	4225x4225
Gauss Jacobi (I)	178	674	1000
Gauss Seidel (I)	92	366	1000
Gauss Jacobi (E)	2,874e-19	9,108e-19	1,247e-09
Gauss Seidel (E)	5,748e-19	4,554e-19	2,371e-15

Considerando que el límite de iteraciones fue de 1000, sólo en las matrices de 289×289 y 1089×1089 se pudo alcanzar la tolerancia esperada. En el caso de la matriz pequeña, ambos métodos pudieron converger en pocas iteraciones, pero Gauss Seidel mostró un mejor rendimiento. Esto se volvió a repetir para la matriz de 1089×1089 . Si bien para la última matriz ambos métodos fueron terminados por la cantidad de iteraciones, el método Seidel estuvo mucho más cerca de converger a la tolerancia esperada respecto al método Jacobi, aunque seidel también demoró más en completar las 1000 iteraciones.

A continuación se muestran los gráficos de resultados y errores de los métodos iterativos:

Para las tres matrices se puede observar un comportamiento similar en cuanto a la aproximación de las soluciones a través de las iteraciones. Ambos métodos alcanzan un valor cercano al final en las primeras iteraciones y con las siguientes simplemente se aumenta la precisión para alcanzar la tolerancia esperada de 10^{-18} .

En cuanto al error, los dos métodos para las tres matrices alcanzan un error cercano a 0 en menos de 20 iteraciones, pero cabe destacar que debido a la tolerancia, el hecho de alcanzar un valor cercano a 0 no es suficiente. Se puede ver el comportamiento del

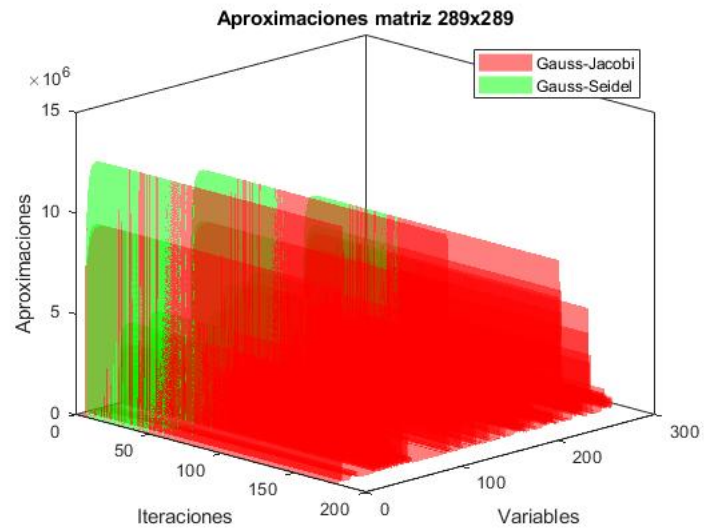


Figura 4.5: Gráficos resultados, matriz 289x289.

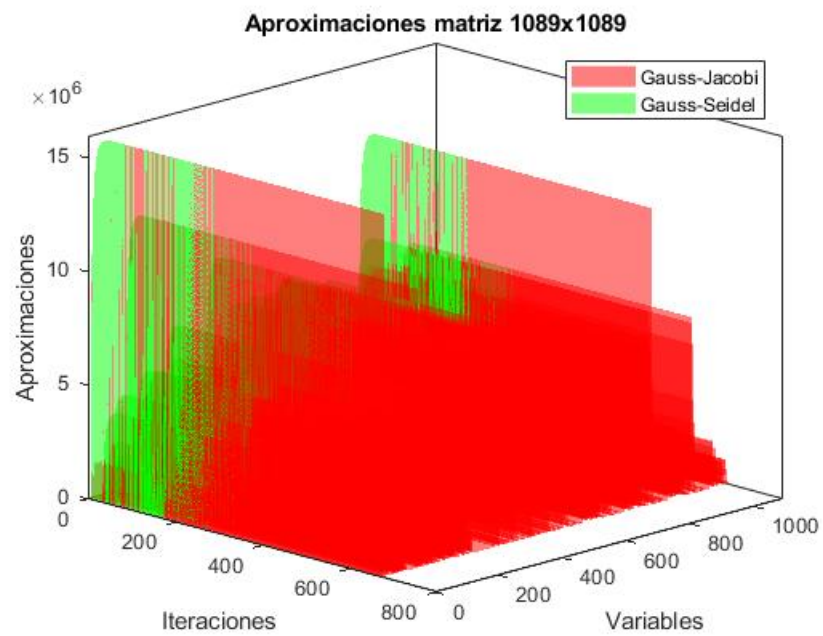


Figura 4.6: Gráficos resultados, matriz 1089x1089.

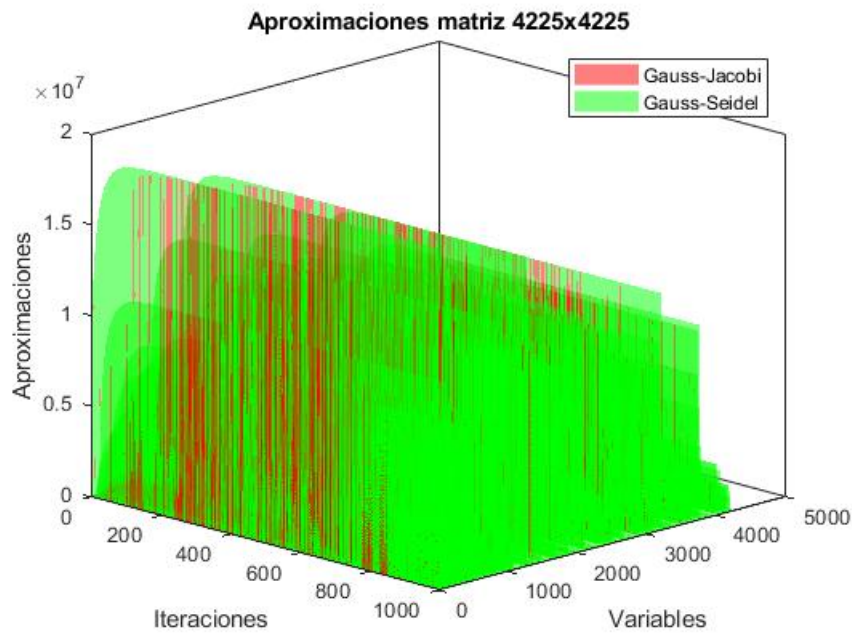


Figura 4.7: Gráficos resultados, matriz 4225x4225.

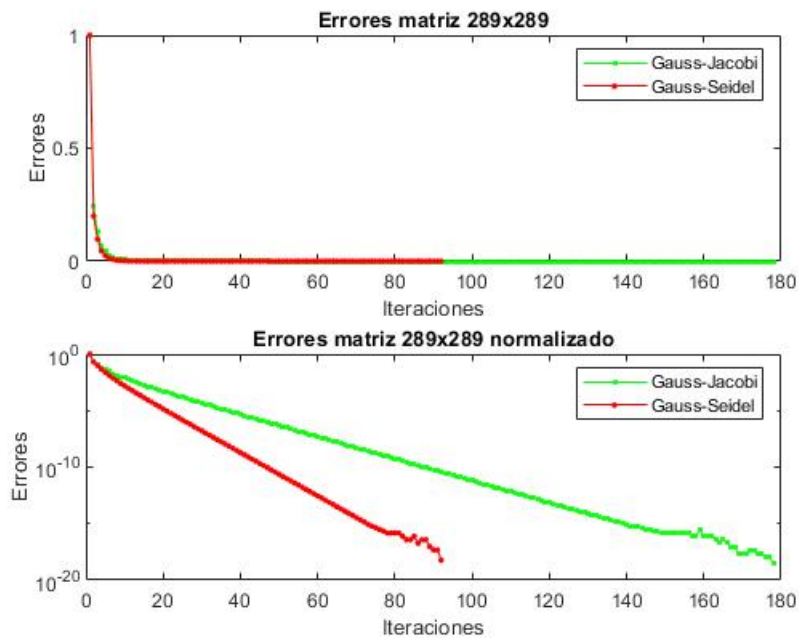


Figura 4.8: Gráficos error, matriz 289x289.

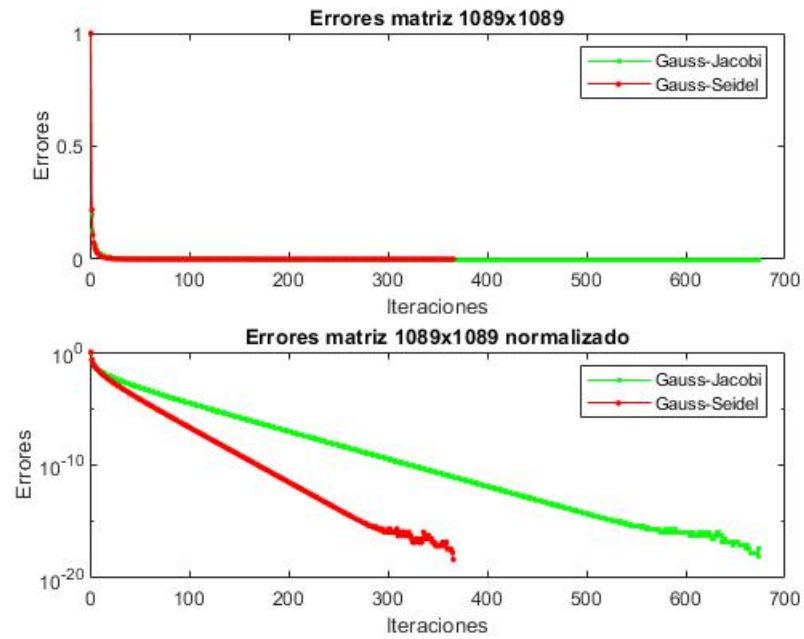


Figura 4.9: Gráficos error, matriz 1089x1089.

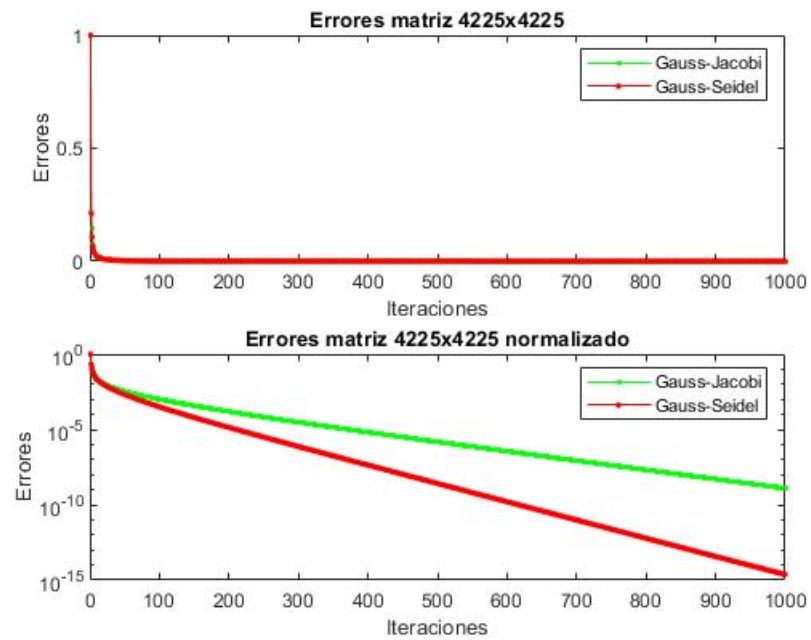


Figura 4.10: Gráficos error, matriz 4225x4225.

error de forma mas fiable en los gráficos semi normalizados logarítmicamente. En dichos gráficos, se pude ver que Gauss Seidel en todos los casos converge mas rápido. Otro punto importante a destacar, consiste en las últimas iteraciones de ambos métodos, como se ve en los gráficos semi normalizados, a diferencia de las primeras iteraciones, el error oscila aumentando y disminuyendo hasta que finalmente alcanza un valor mayor a la tolerancia. Finalmente, Gauss Seidel muestra un mejor comportamiento en cuanto al error y a las aproximaciones para la matriz de 4225×4225 , porque como ya se comprobó con la tabla de número de iteraciones vs error, si bien ambos métodos no alcanzan a converger, Seidel al menos se acerca mucho más a la tolerancia que Gauss Jacobi.

A continuación se muestran los gráficos correspondientes a los resultados de los métodos directos:

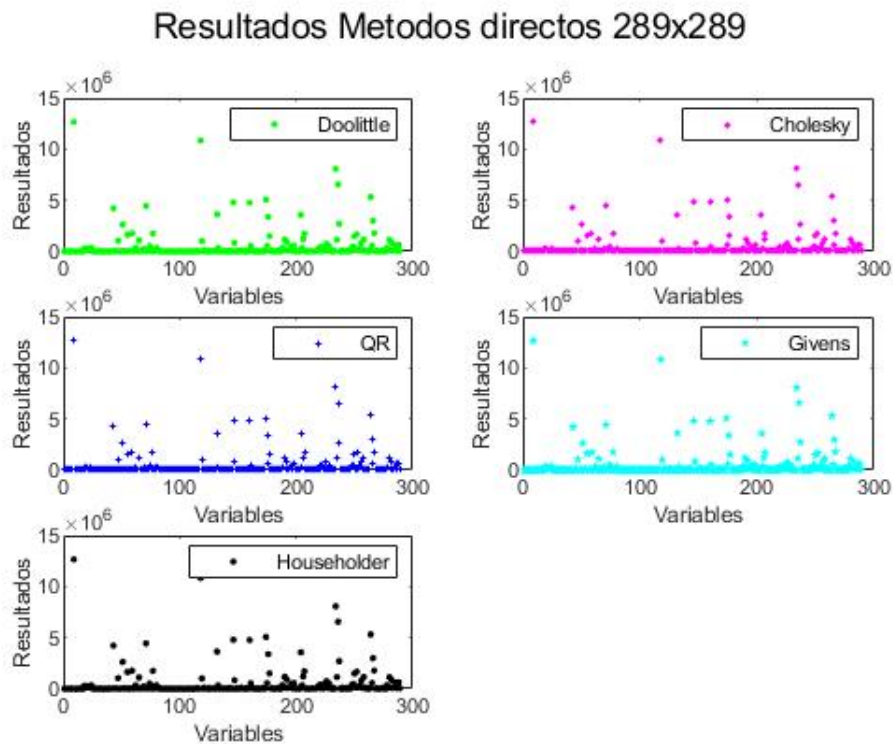


Figura 4.11: Gráficos resultados métodos directos, matriz 289x289.

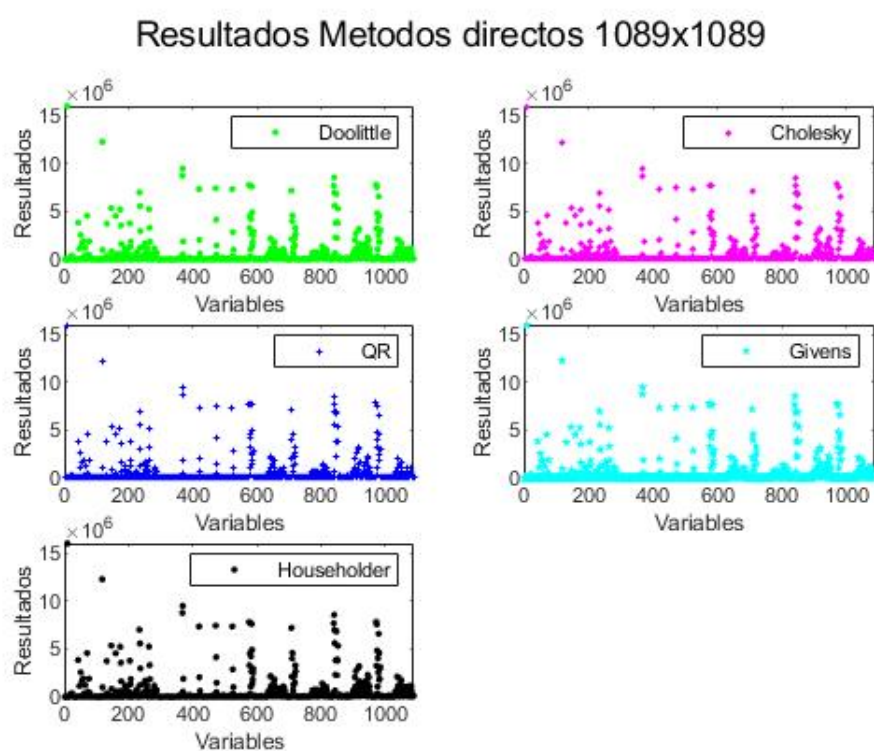


Figura 4.12: Gráficos resultados métodos directos, matriz 1089x1089.

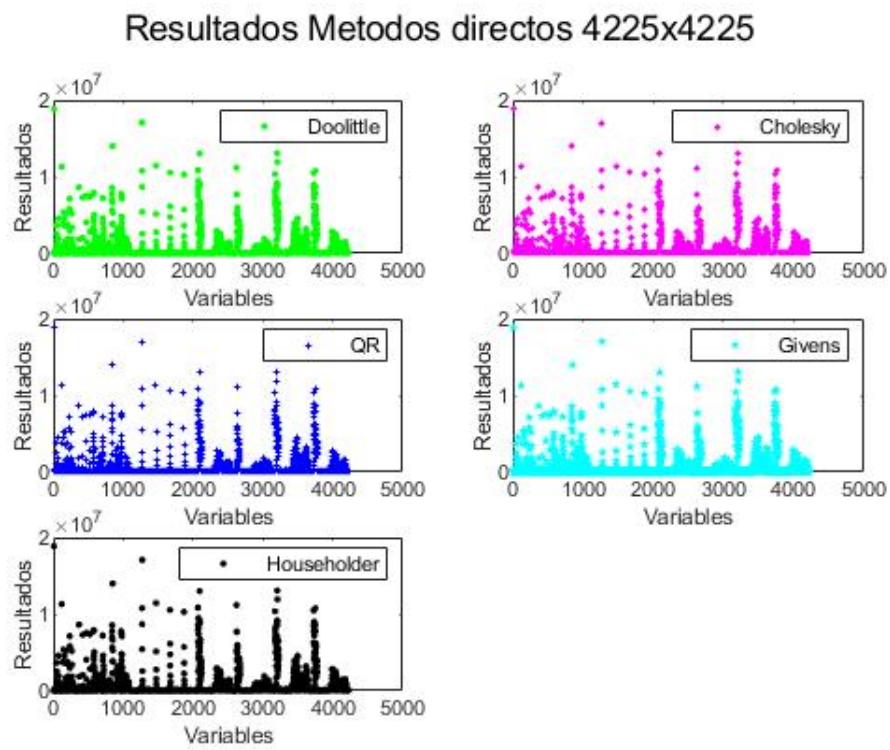


Figura 4.13: Gráficos resultados métodos directos, matriz 4225x4225.

Para el caso de la matriz de 289×289 , todos los métodos presentan los mismos resultados gráficos, todos presentan una concentración de datos en la parte inferior del gráfico, con una serie de outliers distribuidos de la misma forma para todos los métodos.

Para la matriz de 1089×1089 , se tiene un comportamiento similar al caso anterior, todos los métodos muestran resultados similares. La diferencia es que para esta matriz los datos no se concentran tanto en la parte inferior.

Para el caso de la matriz de 4225×4225 se tienen patrones marcados para variables con valores altos y se observa la misma distribución para todos los métodos.

Además, a continuación se muestran los resultados gráficos de las aproximaciones correspondientes a la última iteración de los métodos iterativos:

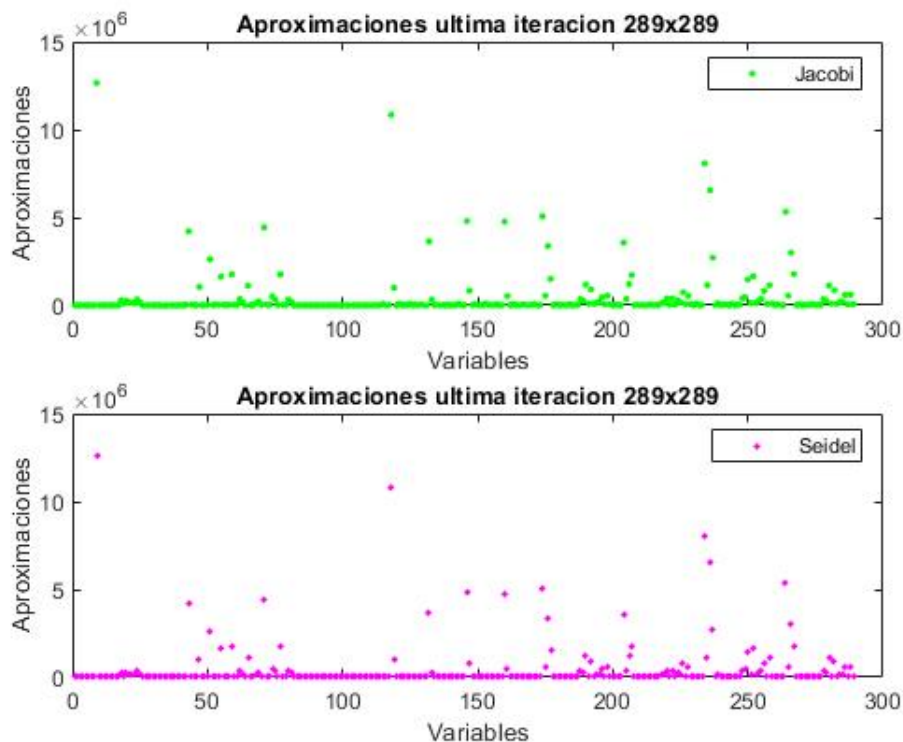


Figura 4.14: Gráficos aproximación 2D métodos iterativos, matriz 289×289 .

Si bien como ya se comprobó en los gráficos 3D, ambos métodos iterativos aproximan

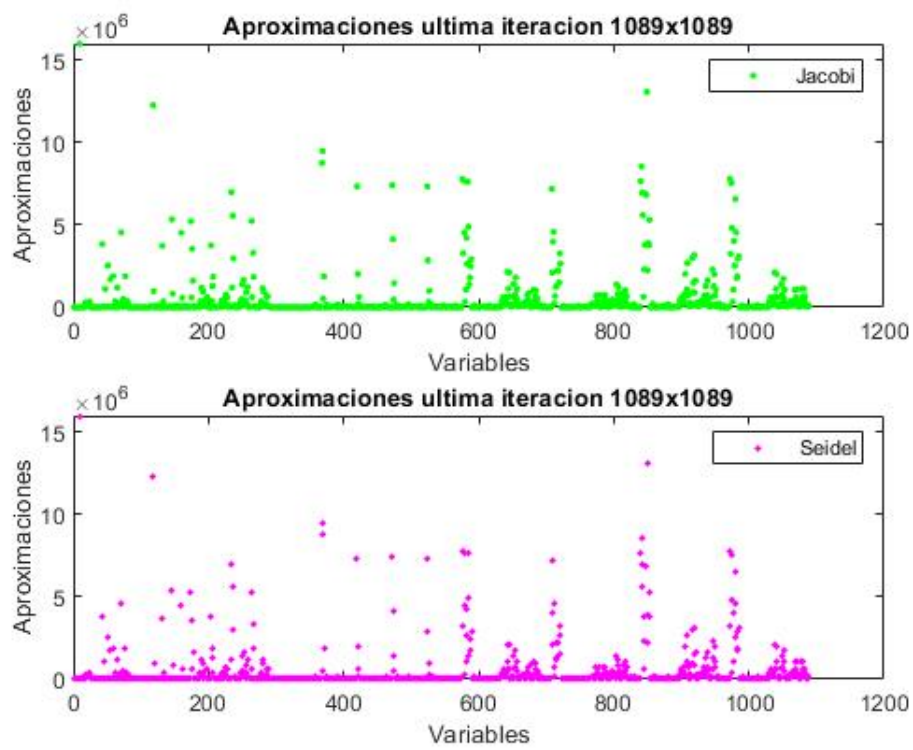


Figura 4.15: Gráficos aproximación 2D métodos iterativos, matriz 1089x1089.

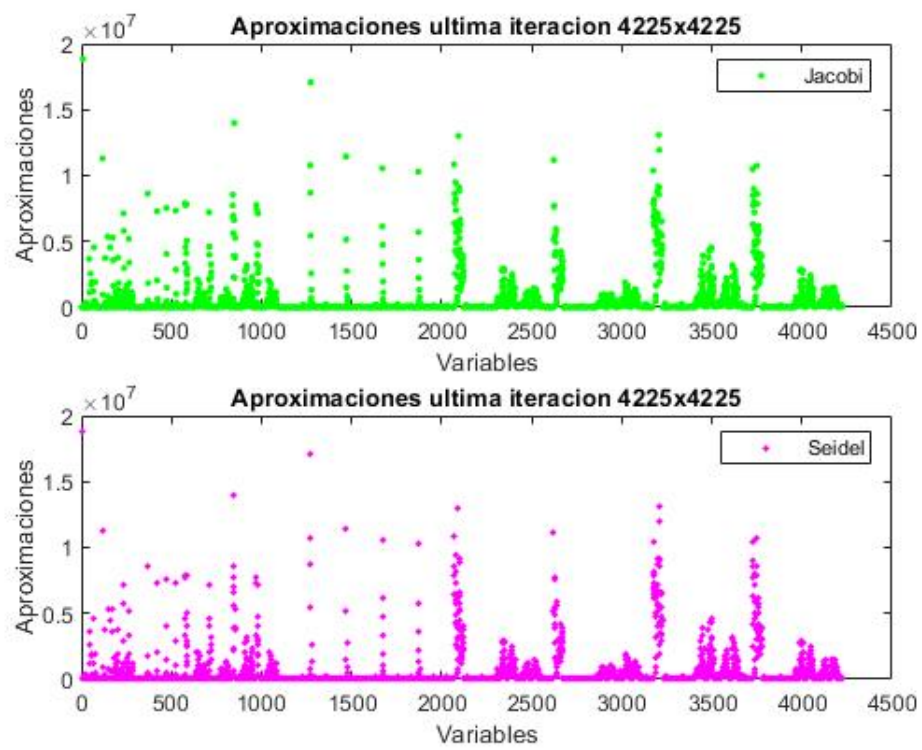


Figura 4.16: Gráficos aproximación 2D métodos iterativos, matriz 4225x4225.

una solución casi idéntica, si se analizan estos gráficos 2D correspondientes a la última aproximación correspondiente a la última iteración de cada método, se puede encontrar una gran similitud (casi idénticos a simple vista) respecto a los resultados obtenidos por los métodos directos.

Finalmente en las siguientes tablas se muestran los errores relativos entre métodos iterativos y métodos directos:

Tabla 4.4: Errores métodos directos vs métodos iterativos

Matrices	Cholesky	Doolittle	Givens	Jacobi	Householder	QR
289x289	1,956e-15	1,985e-15	2,107e-15	7,675e-16	7,096e-15	1,03e-14
1089x1089	1,362e-14	1,363e-14	3,686e-15	3,970e-15	5,503e-14	5,794e-14
4225x4225	6,543e-11	6,543e-11	6,547e-11	6,812e-05	6,568e-11	6,535e-11

El error relativo se calculó comparando la última aproximación del método Gauss Seidel versus todos los resultados de los demás métodos. Según los resultados de la tabla, para la matriz de 289x289 se obtuvo menos error relativo entre todos los métodos y en general se tienen valores muy pequeños, que indican una cantidad mínima de error posiblemente se puede atribuir a los errores asociados con operaciones que utilicen números de punto flotante. El único caso destacable, corresponde a la comparación del resultado de Gauss Seidel vs el resultado de Gauss Jacobi para la matriz de 4225x4225, en este caso el error es mucho mas grande, lo que se debe a que el método Seidel alcanzó a converger mucho más que el método Jacobi en el límite de 1000 iteraciones.

4.3 ANÁLISIS DE APLICACIÓN DEL ALGORITMO DE DECISIÓN

Las tres matrices ya mencionadas fueron analizadas y se determinaron cada una de sus características, las cuales se utilizan en el algoritmo que decide los métodos de solución que serán utilizados para una matriz dada.

Los resultados fueron los siguientes:

Tabla 4.5: Matrices y propiedades

Propiedades	289x289	1089x1089	4225x4225
Cuadrada	X	X	X
Rectangular	-	-	-
Definida positiva	X	-	-
Definida semi positiva	X	X	X
Definida negativa	-	-	-
Diagonal dominante	X	X	X
Simétrica	X	X	X
Dispersa	X	X	X
Grande	-	X	X
Pequeña	X	-	-

Todas las matrices son dispersas, simétricas, definida semi positiva, cuadradas y diagonal dominante. La matriz de 289x289 se considera pequeña, mientras que las matrices de 1089x1089 y 4225x4225 se consideran grandes.

Al aplicar el algoritmo, en base a las propiedades mencionadas, este decide que los siguientes métodos pueden ser ejecutados para cada matriz:

Tabla 4.6: Matrices y métodos

Métodos	289x289	1089x1089	4225x4225
LU Doolittle	X	-	-
Cholesky	X	-	-
QR	-	-	-
Householder	-	-	-
Givens	X	-	-
Gauss Jacobi	X	X	X
Gauss Seidel	X	X	X

Como se puede ver en la tabla Matrices y Métodos, la matriz pequeña de 289x289 será resuelta por 5 métodos, debido a que es la que posee más propiedades de las indicadas en la tabla, lo que permite que más métodos puedan utilizarse para resolverla. La forma en

que decide el algoritmo determina que no todos los métodos se aplicarán a todas las matrices. El algoritmo favorece que se cumpla la mayor cantidad de condiciones. Por ejemplo, si una matriz es pequeña, cuadrada, simétrica, definida positiva, dispersa y diagonal dominante, se le aplicará, Cholesky (por simetría y definida positiva), Seidel y Jacobi (diagonal dominante), LU (pequeña y cuadrada) y givens (pequeña y dispersa). En el caso que una matriz solo tenga menos propiedades importantes como por ejemplo sólo ser simétrica y definida positiva, se resolvería con el método más apropiado, en este caso sería Cholesky. Entonces, la forma en que decide el algoritmo se puede decir que va desde lo más general a lo más específico.

Al aplicar el algoritmo con las respectivas matrices, se tiene el siguiente costo operacional:

Tabla 4.7: Tabla costo operacional según método y matriz, algoritmo

Métodos	289x289	1089x1089	4225x4225
Costo Operacional	50526265	1.2345e+09	3.5600e+10

Estos resultados dependen de la cantidad y tipos de métodos que se utilizaron para cada matriz, por ejemplo para la matriz de 1089×1089 y 4225×4225 solo se utilizaron los métodos iterativos, por lo tanto el algoritmo resulta mucho más barato que ejecutar todos los métodos posibles para ambas matrices. En el caso de la matriz de 289×289 , debido a sus características se le aplicaron los métodos Cholesky, LU Doolittle, Givens, Seidel y Jacobi. Aunque se aplicaron 5 métodos para esta matriz, de todas formas su costo operacional es mucho menor al costo de aplicar Jacobi y Seidel a las dos matrices grandes.

Cabe destacar que si la tolerancia hubiera sido menos estricta y límite de iteraciones hubiera sido mas bajo, la diferencia entre el costo operacional no habría sido tan alta y por lo tanto se hubiera tenido una mejor eficiencia. En el siguiente gráfico se muestran los tiempos de aplicación del algoritmo sobre las tres matrices. Como se puede esperar, la resolución de la matriz de 4225×4225 es el más costoso. Para la matriz de 289×289 los tiempos siguen siendo triviales aunque se sumen los tiempos de los 5 métodos aplicados. Para la matriz de 1089×1089 si bien no se tiene tiempo trivial, pero es mucho menor en

comparación al tiempo ocupado en resolver la matriz más grande. Además se debe considerar que estos tiempos no solo consisten en la suma de los tiempos de los métodos utilizados, sino que también se incluye el tiempo que el algoritmo demoró en determinar cada característica de las matrices. Finalmente, se puede observar que para las matrices

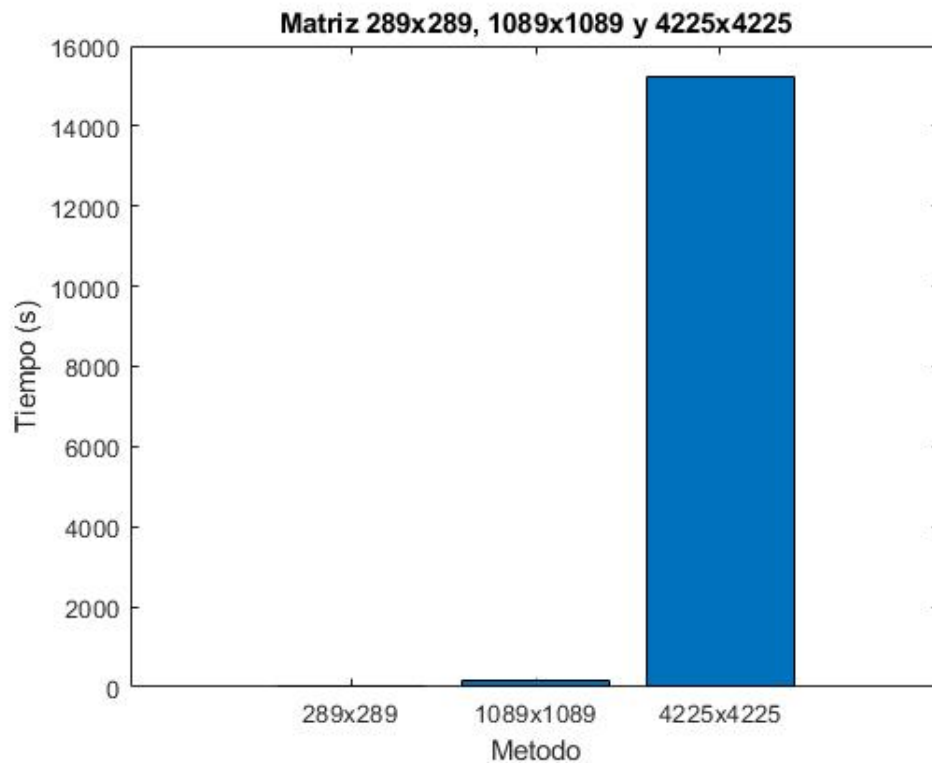


Figura 4.17: Tiempos de aplicación del algoritmo para todas las matrices.

pequeñas la comprobación de las propiedades fue trivial en cuanto al gasto de tiempo, mientras que para la matriz de 4225x4225 se demoró 1500 segundos simplemente en determinar características previas a la utilización de métodos de solución. Aunque esto puede parecer mucho tiempo, de todas formas el gestor es más eficiente que la aplicación de todos los métodos sin comprobación previa, debido a que por ejemplo Householder demoró más de 2000 segundos en resolver la matriz de 4225x4225, mientras que QR, Doolittle y Cholesky estuvieron entre 250 y 500 segundos.

Por lo tanto, se puede decir que la eficiencia del gestor de métodos de solución de sistemas

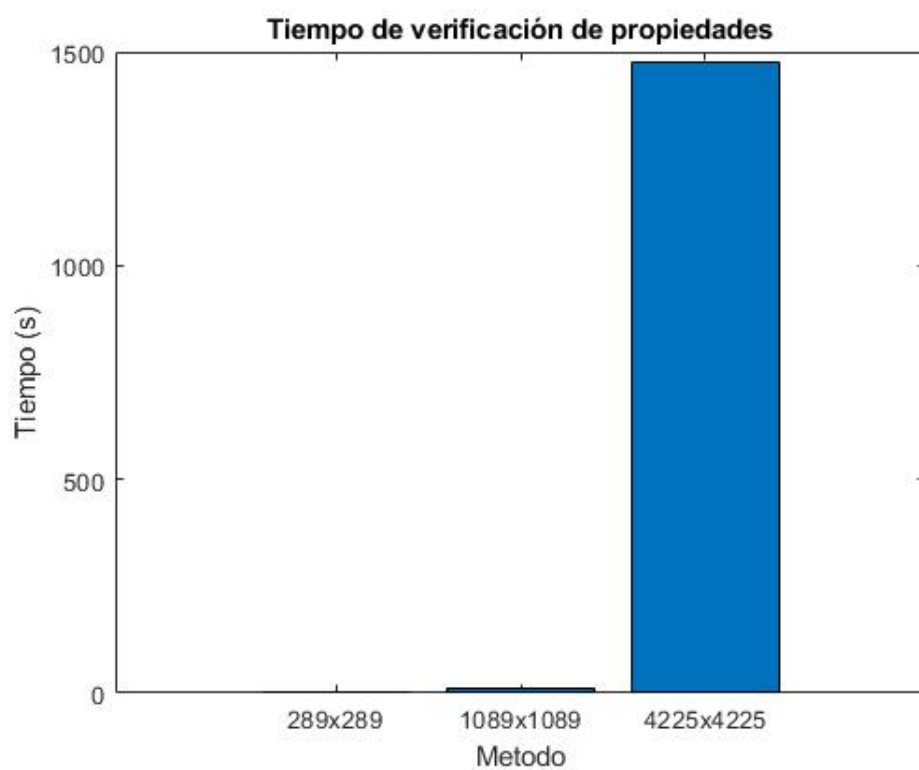


Figura 4.18: Tiempos de verificación de propiedades de matrices.

de ecuaciones lineales es más eficiente que la utilización de todos los métodos sin ningún criterio de por medio, especialmente considerando el caso en que se quisiera resolver una matriz rectangular y no se discriminara entre los métodos a utilizar para la resolución de dicho sistema.

En cuanto a la eficacia, debido a que se intentan utilizar los métodos en las matrices que cumplen la mayor cantidad de propiedades, también se puede decir que el algoritmo favorece una eficacia mayor, debido a que no utilizará métodos en matrices que simplemente no se pueden resolver, que no se puede converger (para métodos directos), asegurando que se obtendrán las mejores soluciones posibles dentro de los métodos disponibles.

CAPÍTULO 5. CONCLUSIONES

Existen métodos de resolución distintos para sistemas de ecuaciones lineales y no lineales. Los sistemas lineales tienen la ventaja de poder ser expresados como matrices y por lo tanto desarrollar métodos utilizando las propiedades y características de estas (dispersa, diagonal dominante, simétrica) a favor y así tener tanto una mejor eficacia (utilización de métodos directos o iterativos con mucha tolerancia) y eficiencia (aplicación de métodos iterativos para matrices más grandes). Con los resultados obtenidos en este laboratorio, se pueden concluir los siguientes puntos:

- Las características de una matriz tienen gran importancia al momento de decidir que método se quiere aplicar, no solo en cuanto a eficiencia y eficacia, sino porque simplemente algunos métodos no se pueden aplicar a algunas matrices, por ejemplo: no se puede asegurar la convergencia para Gauss Jacobi y Seidel si no se cumplen ciertas características. En este laboratorio, se analizaron las matrices con las cuales se debió trabajar y todas cumplían las características mínimas para poder ser aplicadas, pero de todas formas se estudiaron a mayor profundidad para poder desarrollar el algoritmo de decisión.
- Se buscó que el algoritmo resultante priorizara tanto eficacia como eficiencia, debido a que el modo de decisión de métodos iba de lo más general a lo más específico, una matriz puede ser resuelta por varios métodos si es que estos cumplen todas las características necesarias según el algoritmo. Por ejemplo, en las pruebas realizadas la matriz de 289×289 fue resuelta por 5 de los 7 métodos y aún así su costo operacional y tiempo fue trivial. En el caso de las matrices grandes, solo se aplicaron métodos iterativos con una gran tolerancia para beneficiar la eficacia y eficiencia, descartando los métodos directos debido a que sus soluciones podrían no ser lo suficientemente precisas y por lo tanto al aplicarlas se desperdiciaría recursos y tiempo.
- Aunque los métodos directos no aproximan los resultados como los métodos iterativos, de todas formas se producen errores debido a las operaciones de punto flotante

que se deben realizar, por lo tanto no se puede asumir que un método directo tendrá menos error que un método iterativo, esto finalmente dependerá de la cantidad de iteraciones y tolerancia entregadas como condiciones de término del método iterativo. Aunque cabe destacar que con muchas iteraciones y una gran tolerancia, se puede empeorar mucho la eficiencia de un método iterativo en función de mejorar marginalmente su eficacia. Este problema se pudo observar en la resolución de la matriz de 4225×4225 con métodos iterativos, ambos métodos no pudieron alcanzar la tolerancia deseada, el método de Jacobi resultó tener un error bastante grande y el desempeño de tanto Jacobi y Seidel fue mucho peor en cuanto a eficiencia respecto a los métodos directos.

- La resolución de sistemas no lineales también se realiza mediante aproximaciones, lo que genera errores y provoca que la eficiencia y eficacia de los algoritmos (en este caso Newton Raphson) estén dadas por la tolerancia y cantidad de iteraciones máximas. En el caso específico de este laboratorio se alcanzaron valores exactos en la onceava iteración, pero suponiendo que la tolerancia hubiera sido menos estricta y que la cantidad de iteraciones hubiera sido muy pequeña no se habría obtenido una solución buena, por lo tanto la eficacia del método sería muy baja.
- Como ya se mencionó, la tolerancia incide directamente en la eficacia y eficiencia de los métodos iterativos para ecuaciones lineales y no lineales. Una mayor tolerancia aporta eficacia pero reduce la eficiencia. Si tener eficacia sin importar la eficiencia, se puede utilizar una tolerancia que produzca un error de aproximación mínimo. Esto depende directamente de la arquitectura de la máquina (registros), sistema operativo y versión de software (en este caso: Matlab) que se utilice. En el caso de este laboratorio, se pudo trabajar con 64 bits y por esa razón se determinó que la tolerancia debía ser de 10^{-18} si se quería tener un error mínimo.

Además, se logró analizar de forma teórica y práctica los 8 métodos utilizados para poder comprender y comprobar sus diferencias y que todos tienen sus ventajas y desventajas principalmente relacionadas a condiciones de aplicación mínimas y además condiciones

que pueden mejorar o empeorar la eficiencia y eficacia de la solución obtenida. Por ejemplo, si se tiene un problema lineal con muchas variables y la importancia del problema a resolver es mucha porque tiene incidencia con personas en el mundo real, entonces se debería priorizar la eficacia y aplicar el método que produzca la mejor aproximación, aunque su eficiencia no sea la mejor. En cambio si se tiene un problema trivial sin muchas variables y sin incidencias en el mundo real, se puede aplicar un método directo que se demore la menor cantidad de tiempo posible.

Finalmente, se cumplieron los objetivos debido a que se logró implementar en Matlab los 8 métodos propuestos en el enunciado, además del algoritmo de decisión. También se obtuvieron resultados que posteriormente fueron analizados gráficamente y en base a su eficiencia y eficacia.

Bibliografía

- [1] Alvarez, V., Armario, J. A., and Muñoz, F. (n.d.). Sistemas de ecuaciones lineales. http://ma1.eii.us.es/material/Cal_Num_itig_Pres4.pdf.
- [2] Argos, C. G. (2002). Apuntes de metodos numericos 2oe.t.s.i. telecomunicacion universidad de malaga. <http://www.telecos-malaga.com/descargas/apuntes/2Curso/MN/MN-Apuntes.pdf>.
- [3] Biloti, R., Matioli, L., and Yuan, J.-Y. (2007). Generalized givens transformation and its application. https://www.researchgate.net/publication/228964850_Generalized_Givens_Transformation_and_its_Application.
- [4] CCIR/ITESM (2011). Factorizacion qr. <http://cb.mty.itesm.mx/ma1010/materiales/ma1010-23.pdf>.
- [5] Cobos, F. J. (n.d.). Apuntes de algebra numerica para la titulacion de ingenieria informatica. http://www.udesantiagovirtual.cl/moodle/file.php/3415/Teoria_Algebra_Lineal.pdf.
- [6] George, A. (1987). Householder reflections versus givens rotations in sparse orthogonal decomposition. <https://core.ac.uk/download/pdf/81956550.pdf>.
- [7] n.d. (n.d.a). Complexity of algorithms for solution of simultaneous linear equations and matrix multiplication. https://shodhganga.inflibnet.ac.in/bitstream/10603/148188/9/09_chapter203.pdf.

- [8] n.d. (n.d.b). Qr algorithm. <https://www.math.kth.se/na/SF2524/matber15/qrmethod.pdf>.
- [9] Plaza, S. (2007). Métodos numéricos. <http://www.udesantiagovirtual.cl/moodle/file.php/3415/MN2007VF2.pdf>.
- [10] Wikipedia (n.d.a). Lu decomposition. https://en.wikipedia.org/wiki/LU_decomposition.
- [11] Wikipedia (n.d.b). Matriz de diagonal estrictamente dominante. https://es.wikipedia.org/wiki/Matriz_de_diagonal_estrictamente_dominante.