



## Informe de Algoritmos numéricos: Métodos Numéricos

Integrantes: Cristian Espinoza  
Curso: Algoritmos numéricos  
Sección 0-L-1  
Profesor(a): Óscar Rojas Díaz

9 de Mayo de 2018

# Tabla de contenidos

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	1
1.1.1. Objetivos específicos . . . . .	1
1.2. Herramientas . . . . .	2
1.3. Estructura del informe . . . . .	2
<b>2. Descripción de los métodos numéricos</b>	<b>3</b>
2.1. Error . . . . .	3
2.2. Métodos para ecuaciones no lineales . . . . .	4
2.2.1. Método cerrados . . . . .	4
2.2.1.1. Método de Bisección . . . . .	4
2.2.1.2. Método Regular Falsi . . . . .	5
2.2.2. Método abiertos . . . . .	6
2.2.2.1. Método de la Secante . . . . .	6
2.2.2.2. Método de Newton-Raphson . . . . .	6
2.2.2.3. Método de Newton para varias variables . . . . .	7
2.3. Métodos para sistemas de ecuaciones lineales . . . . .	8
2.3.1. Métodos directos . . . . .	9
2.3.1.1. Factorización LU . . . . .	9
2.3.1.2. Método de Crout . . . . .	10
2.3.1.3. Método de Doolitte . . . . .	11
2.3.1.4. Método de Cholesky . . . . .	12
2.3.1.5. Factorización de QR . . . . .	13
2.3.1.6. Método de Givens . . . . .	14
2.3.1.7. Método de Housholder . . . . .	14
2.3.2. Métodos indirectos . . . . .	15
2.3.2.1. Método de Jacobi . . . . .	15
2.3.2.2. Método de Gauss-Seidel . . . . .	16

<b>3. Funciones</b>	<b>18</b>
3.1. Funciones para ecuaciones no lineales . . . . .	18
3.2. Funciones para sistema de ecuaciones lineales . . . . .	20
<b>4. Resultados</b>	<b>22</b>
4.1. Resultado de ecuaciones no lineales . . . . .	22
4.2. Resultado de sistema de ecuaciones no lineal . . . . .	27
4.3. Resultado de sistema de ecuaciones lineales . . . . .	33
<b>5. Análisis de resultados</b>	<b>42</b>
5.1. Métodos no lineales . . . . .	42
5.1.1. Función 1 : $f(x) = 2x - 2^{-x}$ . . . . .	42
5.1.2. Función 2 : $f(x) = e^{-x} - x^2 + 3x - 2$ . . . . .	44
5.1.3. Función 3 : $f(x) = x^3 - 7x^2 + 14\sin(2 * \pi * x) - 6$ . . . . .	46
5.2. Sistema de ecuaciones no lineales . . . . .	48
5.2.1. Sistema de ecuaciones 1 : . . . . .	48
5.2.2. Sistema de ecuaciones 2 : . . . . .	48
5.3. Métodos lineales . . . . .	49
5.3.1. Matriz 289x289 : . . . . .	49
5.3.2. Matriz 1089x1089 : . . . . .	52
5.3.3. Matriz 4225x4225 : . . . . .	54
<b>6. Conclusiones</b>	<b>56</b>
<b>Bibliografía</b>	<b>58</b>

# Índice de figuras

1.	Resultado de las aproximaciones de la función 1 . . . . .	23
2.	Resultados de los errores obtenidos en la función 1 . . . . .	23
3.	Resultado de las aproximaciones de la función 1 . . . . .	24
4.	Resultados de los errores obtenidos en la función 1 . . . . .	25
5.	Resultado de las aproximaciones de la función 3 . . . . .	26
6.	Resultados de los errores obtenidos en la función 3 . . . . .	26
7.	Resultado de las aproximaciones de la función 1 de sistema de ecuaciones no lineal . . . . .	28
8.	Resultados de los errores obtenidos en la función 1 de sistema de ecuaciones no lineal . . . . .	29
9.	Resultado de las aproximaciones de la función 2 de sistema de ecuaciones no lineal . . . . .	31
10.	Resultados de los errores obtenidos en la función 2 de sistema de ecuaciones no lineal . . . . .	32
11.	Apreciación en formato 3D de la función 2 de sistema de ecuaciones no lineal	32
12.	Aproximación de la matriz 289x289 por métodos iterativos . . . . .	33
13.	Aproximación de la matriz 289x289 por métodos directos . . . . .	34
14.	Error obtenido en el desarrollo de la matriz de 289x289 . . . . .	34
15.	Aproximación de la matriz 1089x1089 por métodos iterativos . . . . .	36
16.	Aproximación de la matriz 1089x1089 por métodos directos . . . . .	37
17.	Error obtenido en el desarrollo de la matriz de 1089x1089 . . . . .	37
18.	Aproximación de la matriz 4225x4225 por métodos iterativos . . . . .	39
19.	Aproximación de la matriz 4225x4225 por métodos directos . . . . .	40
20.	Error obtenido en el desarrollo de la matriz de 4225x4225 . . . . .	40

# Índice de cuadros

1.	Tabla de valores iniciales para la función (1) . . . . .	19
2.	Tabla de valores iniciales para la función (2) . . . . .	19
3.	Tabla de valores iniciales para la función (3) . . . . .	19
4.	Tabla de resultados para la función (1) . . . . .	22
5.	Tabla de resultados para la función (2) . . . . .	24
6.	Tabla de resultados para la función (3) . . . . .	25
7.	Tabla de aproximaciones para sistema de ecuación no lineal - Función 1 . . .	27
8.	Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 1 . . . . .	27
9.	Tabla de errores obtenidos para sistema de ecuaciones no lineal - Función 1 .	27
10.	Tabla de aproximaciones para sistema de ecuación no lineal - Función 2 . . .	29
11.	Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 2 . . . . .	30
12.	Resultados de los errores obtenidos en la función 1 de sistema de ecuaciones no lineal . . . . .	30
13.	Tabla de resumen para matriz de 289x289. . . . .	35
14.	Tabla de de costos espaciales para matriz de 289x289 . . . . .	35
15.	Tabla de resumen para matriz de 1089x1089. . . . .	38
16.	Tabla de de costos espaciales para matriz de 289x289 . . . . .	38
17.	Tabla de resumen para matriz 4225x4225. . . . .	41
18.	Tabla de de costos espaciales para matriz de 289x289 . . . . .	41
19.	Tabla de mayor iteración . . . . .	42
20.	Tabla de mayor iteración . . . . .	43
21.	Tabla de mayor iteración . . . . .	43
22.	Tabla de mayor iteración . . . . .	44
23.	Tabla de mayor iteración . . . . .	45
24.	Tabla de mayor iteración . . . . .	45
25.	Tabla de mayor iteración . . . . .	46

26.	Tabla de mayor iteración . . . . .	47
27.	Tabla de mayor iteración . . . . .	47
28.	Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 1 . . . . .	48
29.	Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 2 . . . . .	49
30.	Tabla de resumen tiempo de ejecución para sistema 289x289. . . . .	50
31.	Tabla de de costos espaciales para matriz de 289x289 . . . . .	50
32.	Tabla de resumen tiempo de ejecución para sistema 1089x1089. . . . .	52
33.	Tabla de de costos espaciales para matriz de 1089x1089 . . . . .	53
34.	Tabla de resumen tiempo de ejecución para sistema 1089x1089. . . . .	54
35.	Tabla de de costos espaciales para matriz de 4225x4225 . . . . .	55

# 1. Introducción

Por como bien se sabe actualmente todo gira en torno a los números y diversos cálculos matemáticos, pero para esto se usan distintos tipos de operaciones. Debido a lo anterior, es que se usa una rama de la matemática llamada análisis numérico, que mediante distintos tipos de métodos numéricos se realizan cálculos y a su vez un completo análisis de esos resultados obtenidos.

Cabe destacar que el uso de métodos numéricos otorga una infinidad de habilidades, las cuales pueden ser ocupadas en el día a día. Sin embargo, tenemos que la computadora no puede aprender ciertas habilidades, por el simple motivo que no es un ser pensante. Debido a esto es que se ocupan técnicas para disminuir la dificultad de los distintos métodos numéricos, algunas de las mas utilizadas es la aproximación.

El desarrollo del presente laboratorio se divide en dos partes las cuales son categorizadas como: aplicación de métodos para la solución de ecuaciones no lineales y lo mismo para sistemas de ecuaciones lineales.

## 1.1. Objetivos

El objetivo que se busca en dicho laboratorio, es realizar cada uno de los métodos solicitados en su totalidad, además de analizar cada uno de los resultados obtenidos.

### 1.1.1. Objetivos específicos

1. Implementar los métodos de sistema de ecuaciones no lineales : Bisección, Secante, Regular Falsi, Newton-Raphson, Newton-Raphson Multivariable.
2. Realizar un análisis a 3 funciones declaradas en el enunciado con todos los métodos de sistema de ecuaciones no lineales.
3. Implementar los métodos de sistema de ecuaciones lineales : Jacobi, Gauss-Seidel, LU, Cholesky, QR, Givens y Householder.
4. Realizar un análisis a 3 matrices de distintas dimensiones con los métodos de sistemas de ecuaciones lineales.

## **1.2. Herramientas**

Las herramientas ocupadas para el desarrollo del informe, fueron las siguientes:

1. Para la realización del código fuente se ocupó Matlab R2017a.
2. Para la realización de la documentación del laboratorio se ocupó Latex - Sharlatex.

## **1.3. Estructura del informe**

Este informe consta de un capítulo donde se procede a describir cada uno de los métodos que se van a utilizar dentro del desarrollo del laboratorio número 1. Luego, se procede a dar una pequeña explicación de lo que se obtendrá en el laboratorio definiendo parámetros de trabajos para el transcurso del mismo. Posteriormente, se muestran los resultados de cada uno de los métodos realizados, además de sus respectivas tablas de datos y gráficos obtenidos. Por último, se realiza un análisis de cada uno de ellos para luego pasar al último capítulo del informe donde se concluye el trabajo realizado.



## 2. Descripción de los métodos numéricos

En este capítulo se procede a describir cada uno de los métodos que vamos a utilizar en el desarrollo del laboratorio.

Principalmente los métodos se basan en encontrar la raíz mas exacta de una determinada función.

### 2.1. Error

En su mayoría las distintos procedimientos que se realizan con numero reales es imposible representarlos en su forma exacta.

Por lo cual tenemos que calcular el error asociado a cada uno de estos procedimientos, de modo que es expresado como  $E(x_a)$ . Lo anterior nos representa el error aproximado por un numero  $x_a$  con respecto al valor real que se tiene de  $x_b$ .

$$E(x_a) = x_a - x_b \quad (1)$$

Existen casos en donde el valor que tiene  $x_b$  es mayor que  $x_a$ , lo que conlleva a tener un error negativo, para evitar este tipo de inconvenientes se utiliza "Valor absoluto" de la ecuación. Además, tenemos que para eventualidades de estudios se utiliza el error porcentual.

$$|E(x_a)| = \frac{|x_a - x_b|}{|x_a|} \quad (2)$$

Además, se define el costo espacial  $C_{espacial}$ , el cual se calcula aumentando un contador cada vez que se produce una operación matemática básica , ya sea: suma, resta, multiplicación y división, quedando su formula como:

$$C_{espacial} = \sum_{i=1}^n x_+ + x_- + x_* + x_ / \quad (3)$$

Donde tenemos que:

- $x_+$ : Representa la cantidad de sumas que existen.

- $x_-$ : Representa la cantidad de restas que existen.
- $x_+$ : Representa la cantidad de divisiones que existen.
- $x_*$ : Representa la cantidad de multiplicaciones que existen.

## 2.2. Métodos para ecuaciones no lineales

### 2.2.1. Método cerrados

Acá explicaremos los métodos que necesitan de dos puntos para poder ser utilizados, es decir, que contiene un punto de inicio y un punto de termino.

#### 2.2.1.1. Método de Bisección

Este método esta dentro de la categoría de métodos cerrados, por el simple hecho que necesita dos puntos para poder ser aplicado.

Ahora tenemos que este método ocupa para su desarrollo el teorema del valor medio el cual procederemos a explicarlo a continuación.

#### **Teorema del valor medio**

Sea una función definida como  $f(x)$  continua en un intervalo definido en  $[a, b]$ .

En el caso que  $f(a)$  y  $f(b)$  tengan signos distinto nos indica que existe algún  $r$  entre el intervalo  $[a, b]$  para el cual  $f(r) = 0$ .

De modo que aseguramos que  $f(a) * f(b) < 0$  podemos decir que  $f(x)$  tiene a lo menos una raíz dentro del intervalo  $[a, b]$ .

Ahora volviendo al procedimiento del método, tenemos que al tener el intervalo definido que cumpla con el teorema del valor medio con respecto a la función  $f(x)$ . Luego, procedemos a calcular la semi suma entre  $a$  y  $b$ , la cual queda definida como :

$$m = \frac{a + b}{2} \quad (4)$$

Luego tenemos dos intervalo candidatos, los cuales son:  $[a, m]$  y  $[m, b]$ , de estos anteriores tenemos que escoger al que cumpla con las condiciones del teorema del valor medio.

Tener en cuenta que podemos repetir el procedimiento las veces que estimemos conveniente, a medida que mas veces se realiza el procedimiento el valor de  $m$  se acerca mas al verdadero valor de la raíz de la función  $f(x)$ .

#### 2.2.1.2. Método Regular Falsi

Tenemos que este método al igual que Bisección necesita de dos puntos para poder ser aplicado.

EL primer paso que realizamos es escoger los valores iniciales  $x_0$  y  $x_1$ , tal que cumplan con el teorema del valor medio con respecto a la función  $f(x)$ .

La aproximación de  $x_2$  se realiza con la siguiente formula :

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)} \quad (5)$$

Para poder determinar con que puntos calcularemos  $x_3$  debemos verificar el signo que nos entrega  $f(x_2) * f(x_3)$ . En este caso tenemos dos opciones, las cuales son:

- $f(x_2) * f(x_3) < 0$  : Tenemos que la raíz la tenemos contenidos dentro del intervalo  $[x_1, x_2]$ , lo cual implica que  $x_3$  es la solución de la recta que conecta a los puntos  $(x_1, f(x_1))$  y  $(x_2, f(x_2))$ .
- $f(x_2) * f(x_3) > 0$  : Tenemos que la raíz la tenemos contenidos dentro del intervalo  $[x_0, x_2]$ , lo cual implica que  $x_3$  es la solución de la recta que conecta a los puntos  $(x_0, f(x_0))$  y  $(x_2, f(x_2))$ .

Luego tenemos que para volver a iterar solo nos basta con aplicar la formula :

$$x_r = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, k \geq 1 \quad (6)$$

Donde tenemos que  $x_r$ , nos representa la iteración que estamos calculando y  $k$ , es la iteración que estamos actualmente.

### 2.2.2. Método abiertos

Acá explicaremos los métodos que necesitan de solamente un punto para poder ser utilizados, es decir, que solamente tiene un punto de inicio.

#### 2.2.2.1. Método de la Secante

Este método como se menciono anteriormente necesita solamente de un punto para poder ser aplicado.

El método de la secante soluciona uno de los grandes problemas que tiene el método "Newton-Raphson" que es calcular la  $f'(x)$ . Por el motivo, que a veces el costo operacional para poder calcular la derivada de la función  $f'(x)$  es alto, por el motivo mencionado anteriormente es que en dichos casos se prefiere ocupar el método de la "Secante".

El método de la Secante aproxima la pendiente de la recta que une a la función  $f(x)$  evaluada en el punto  $x_{n+1}$  y en el punto anterior  $x_n$ , es decir, que este método hace uso de la iteración anterior para trazar las lineas secantes y así aproximarse de manera mas certera a la raíz de la función  $f(x)$ .

La ecuación del método es:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, n \geq 1 \quad (7)$$

#### 2.2.2.2. Método de Newton-Raphson

Este método como se menciono anteriormente necesita solamente de un punto para poder ser aplicado.

Tener en cuenta que este método es uno de los mas utilizados para aproximar la solución de una función  $f(x)$ . El modelo de uso es ir iterando con el punto que recibe como parámetro con la siguiente formula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n \geq 1 \quad (8)$$

### 2.2.2.3. Método de Newton para varias variables

Este método busca las raíces de un sistema de ecuaciones del tipo como:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = f_1(x) = 0 \\ f_2(x_1, x_2, \dots, x_n) = f_2(x) = 0 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ f_n(x_1, x_2, \dots, x_n) = f_n(x) = 0 \end{cases}$$

Donde  $x^k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T$  es un vector de n-dimensiones y  $k$  significa el número de iteración que se ha llevado a cabo. Primero se resuelve el sistema de ecuaciones lineales:

$$J_k(f_1, f_2, \dots, f_n) \begin{pmatrix} h_1^{(k)} \\ h_2^{(k)} \\ \dots \\ \dots \\ kh_n^{(k)} \end{pmatrix} = - \begin{pmatrix} f_1(x) \\ f_2(x) \\ \dots \\ \dots \\ f_n(x) \end{pmatrix}$$

Para  $H^k = (h_1^{(k)}, h_2^{(k)}, \dots, h_n^{(k)})$ , donde  $J_k(f_1, f_2, \dots, f_n)$  denota el jacobiano de  $F = (f_1, f_2, \dots, f_n)$  evaluado en el punto  $x^k$ . Posterior a esto, se realiza la iteración:

$$x^{k+1} = x^k + H^{(k)} \quad (9)$$

Expresado de forma matricial queda:

$$\begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{pmatrix} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} + J_k^{-1}(f_1, f_2, \dots, f_n) \begin{pmatrix} f_1(x^{(k)}) \\ f_2(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{pmatrix}$$

Así sucesivamente se sigue iterando hasta satisfacer alguna condición de parada.

## 2.3. Métodos para sistemas de ecuaciones lineales

Estos métodos se especializan en desarrollar sistema de ecuaciones tales que son representadas como:

$$Ax = b \rightarrow x = bA^{-1} \quad (10)$$

Donde tenemos que:

■ A:

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix}$$

■ b:

$$\begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{nn} \end{pmatrix}$$

■  $A, b \in R^n$

■ x:

$$\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{nn} \end{pmatrix}$$

■  $x \in R^n$  y es incógnita.

Además, los métodos que se tiene par obtener el valor de  $x$  se clasifican como : métodos directos e indirectos.

### 2.3.1. Métodos directos

El planteamiento que estos métodos toman es tomar el sistema de ecuaciones inicial y reducirlo a un nuevo sistema de ecuaciones mas fácil de resolver.

#### 2.3.1.1. Factorización LU

Esta es una forma de factorizar una matriz, como el producto de una matriz triangular inferior  $L$ , con una matriz triangular superior  $U$ . Mediante el sistema de ecuaciones ya definido como :

$$Ax = b \quad (11)$$

Donde  $A$  se puede descomponer de la siguiente forma :

$$A = LU \quad (12)$$

Si la descomposición anterior la expresamos en matrices, tenemos lo siguiente:

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

Donde reemplazando en la formula anterior dada, nos queda de la siguiente forma:

$$LUx = b \quad (13)$$

Esta descomposición nos sirve para poder resolver sistemas de ecuaciones lineales, a través de distintos métodos, entre ellos tenemos: Crout , Doolitte y Choslesky, los cuales veremos a continuación.

### 2.3.1.2. Método de Crout

Este método aplica la facturación LU anteriormente mencionada, en donde la matriz U en su diagonal contiene solamente 1. De manera que comienza con resolverse el siguiente sistema de ecuaciones:

$$LUx = b \quad (14)$$

Donde se produce un cambio de variable  $y = Lx$ , y se obtiene las soluciones de  $y$ , quedando de la siguiente manera:

$$Uy = b \quad (15)$$

$$\begin{pmatrix} 1 & U_{12} & \cdots & U_{1n} \\ 0 & 1 & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} * \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{n1} \end{pmatrix} = \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{pmatrix}$$

Finalmente, para obtener las raíces de  $x$ , se resuelve el siguiente sistema de ecuaciones :

$$Lx = y \quad (16)$$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ L_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & 1 \end{pmatrix} * \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{n1} \end{pmatrix}$$

Donde obtenemos como resultado final los valores de  $x$ , expresados de la siguiente forma:



$$\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{nn} \end{pmatrix}$$

### 2.3.1.3. Método de Doolitte

Este método aplica la facturación LU anteriormente mencionada, en donde la matriz L en su diagonal contiene solamente 1. De manera que comienza con resolverse el siguiente sistema de ecuaciones:

$$LUx = b \quad (17)$$

Donde se produce un cambio de variable  $y = Ux$ , y se obtiene las soluciones de  $y$ , quedando de la siguiente manera:

$$Ly = b \quad (18)$$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ L_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & 1 \end{pmatrix} * \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{n1} \end{pmatrix} = \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{pmatrix}$$

Finalmente, para obtener las raíces de  $x$ , se resuelve el siguiente sistema de ecuaciones :

$$Ux = y \quad (19)$$

$$\begin{pmatrix} 1 & U_{12} & \cdots & U_{1n} \\ 0 & 1 & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} * \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{n1} \end{pmatrix}$$

Donde obtenemos como resultado final los valores de  $x$ , expresados de la siguiente forma:

$$\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{nn} \end{pmatrix}$$

#### 2.3.1.4. Método de Cholesky

Este método es otro de los que ocupa la factorización de LU, además de tener una restricción adicional la cual recae en  $\det(A) \neq 0$ .

Sea la función:

$$Ax = b \tag{20}$$

La función anterior la descomponemos de la siguiente manera:

$$A = LL^T \tag{21}$$

Este método al igual que los anteriores sigue el mismo procedimiento, donde hace un cambio de variable del modo  $y = L^T x$ :

$$Ly = b \tag{22}$$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ L_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & 1 \end{pmatrix} * \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{n1} \end{pmatrix} = \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{pmatrix}$$

Finalmente para obtener las soluciones de  $x$ , tenemos que:

$$L^T x = y \tag{23}$$

$$\begin{pmatrix} 1 & L_{12} & \cdots & L_{1n} \\ 0 & 1 & \cdots & L_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} * \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{pmatrix}$$

Donde obtenemos como resultado final los valores de  $x$ , expresados de la siguiente forma:

$$\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{nn} \end{pmatrix}$$

#### 2.3.1.5. Factorización de QR

Dada una matriz  $A$ , la cual no tiene que ser necesariamente cuadrada, con columnas linealmente independientes, encontraremos matrices  $Q$  y  $R$ , las cuales:

- $A = QR$
- Las columnas de  $Q$  son ortonormales.
- $Q$  es del mismo tamaño que  $A$ .
- $R$  es triangular superior.

Al cumplir estas condiciones, se aplica el teorema de Gram-Schmidt a las columnas de  $A$ . Recordar que el teorema de Gram-Schmidt dice que a partir de los vectores linealmente independientes  $v_1, v_2, \dots, v_n$ , se cumple:

$$u_1 = v_1 \quad (24)$$

$$u_k = v_k - \sum_{j=1}^{k-1} \frac{\langle v_k, u_j \rangle}{\|u_j\|^2} u_j, j = 2, 3, \dots, k \quad (25)$$

Los vectores  $u_1, u_2, \dots, u_k$  son ortogonales.

#### 2.3.1.6. Método de Givens

Se parte por definir una matriz de manera:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \cos(x) & \cdots & \sin(x) & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \sin(x) & \cdots & \cos(x) & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

El método de Givens consiste en lo siguiente : En sucesivas transformaciones de rotación usando matrices de Givens hasta reducir la matriz  $A$  rectangular  $m \times n$  a una triangular. Si  $m \geq n$ :

$$G_n^t \dots G_1^1 * A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, A \in R^{n \times n} R^1 \in R^{n \times n} \quad (26)$$

Siendo  $R_1$  triangular superior.

La matriz  $Q = G_1 \cdot \dots \cdot G_n$  no se suele calcular y sólo se almacena un número por rotación en la posición anulada. Se suele usar este método con matrices dispersas

#### 2.3.1.7. Método de Housholder

Este método se define por la siguiente formula :

$$H = I - \frac{2}{v'v} * vv^t \quad (27)$$

Donde  $v$  es un vector no nulo, de forma que el producto  $v * v^t$  es una matriz cuadrada (simétrica  $n \times n$  y de rango 1) y  $v^t * v$  un número.

El método de Householder consiste en lo siguiente : En la sucesiva realización de transformaciones (un total de  $n$ ) de reflexión utilizando matrices de Householder hasta reducir  $A \in R^{m \times n}$  a una matriz triangular. Si  $m \geq n$ :

$$H_n^t \dots H_1^t * A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, R_1 \in R^{n \times n} \text{ triangular superior} \quad (28)$$

La diferencia con el método de Givens es que la matriz  $H$ , al pre-multiplicar a  $A$ , hace que se anulen todos los elementos de la columna que necesite, no elemento a elemento.

### 2.3.2. Métodos indirectos

En esta sección tendremos a los métodos de Jacobi y Gauss-Seidel.

#### 2.3.2.1. Método de Jacobi

Dado un sistema de ecuaciones de la forma :

$$Ax = B \quad (29)$$

Se descompone la matriz de  $A$ , de tal forma :

$$A = D + R \quad (30)$$

Donde  $D$  es una matriz diagonal y  $R = L + U$ , con  $L$  la matriz triangular inferior y  $U$  la matriz triangular superior. Luego procedemos a reemplazar y obtenemos lo siguiente:

$$Dx + Rx = b \quad (31)$$

Luego para obtener la solución tenemos que resolver la siguiente ecuación, la cual nos entregará la solución del sistema de ecuaciones

$$x = D^{-1}b - D^{-1}Rx \quad (32)$$

Donde tenemos que si:  $a_{ii} \neq 0 \forall i$ , por regla de la iteración de Jacobi, tenemos que la formula queda expresada como :

$$x_i^{k+1} = D^{-1}b - D^{-1}Rx^k \quad (33)$$

Donde tenemos que  $k$ , nos representa la iteración que estamos realizando, entonces nuestra fomrula nos queda como:

$$x_i^{k+1} = \frac{1}{a_{ii}}(b_i - \sum_{i \neq j} r_{ij}x_j^k, i = 1, 2, 3, 4, 5, \dots) \quad (34)$$

#### 2.3.2.2. Método de Gauss-Seidel

Este método de encarga de resolver sistema de ecuaciones, que viene expresados de la siguiente forma:

$$Ax = b \quad (35)$$

Donde procedemos a descomponer la matriz de A, quedando como:

$$R = A - Q \quad (36)$$

En este caso, tenemos que  $Q$  es una matriz triangular inferior, incluyendo su diagonal, donde la formula cambia y queda como :

$$Qx^k = b - Rx^{k+1} \quad (37)$$

Donde finalmente tenemos que la formula nos va a quedar de la siguiente manera:

$$Qx^k = \frac{(b_i - \sum_{j \neq i}^{i-1} a_{ij}x_j^k - \sum_{j \neq i+1}^n a_{ij}x_j^{k-1})}{a_{ii}}, i = 1, 2, 3, 4, 5, \dots \quad (38)$$

### 3. Funciones

#### 3.1. Funciones para ecuaciones no lineales

Las funciones para la primera parte del laboratorio son las siguientes :

- Función 1:

$$x - 2^{-x} \quad (39)$$

- Función 2:

$$e^{-x} - x^2 + 3x - 2 \quad (40)$$

- Función 3:

$$x^3 - 7x^2 + 14\sin(2 * \pi * x) - 6 \quad (41)$$

Para estas funciones declaradas anteriormente, le aplicaremos los siguientes métodos :

- Bisección
- Secante
- Regular Falsi
- Newton-Raphson

Primero declaramos que la tolerancia que se ocupara para la ejecución de cada una de las funciones recae en :

$$E = 10^{-18} \quad (42)$$

Los puntos iniciales escogidos para cada una de las funciones se pueden observar en la Cuadro 1, Cuadro 2 y Cuadro 3. Los cuales fueron escogidos con la regla de fourier



$f(x) = x - 2^{-x}$		
<b>Método</b>	$x_0$	$x_1$
Bisección	0	2
Secante	0	2
Regular Falsi	0	2
Newton-Raphson	0	-

Cuadro 1: Tabla de valores iniciales para la función (1)

$f(x) = e^{-x} - x^2 + 3x - 2$		
<b>Método</b>	$x_0$	$x_1$
Bisección	0	1
Secante	0	1
Regular Falsi	0	1
Newton-Raphson	0	-

Cuadro 2: Tabla de valores iniciales para la función (2)

$f(x) = x^3 - 7x^2 + 14\sin(2 * \pi * x) - 6$		
<b>Método</b>	$x_0$	$x_1$
Bisección	0	0.3
Secante	0	0.3
Regular Falsi	0	0.3
Newton-Raphson	0	-

Cuadro 3: Tabla de valores iniciales para la función (3)

Para las funciones con las cuales se probó el método de Newton-Raphson Multi-variable fueron dos, las cuales son :

- Función 1 :

$$\begin{cases} x^2 - 10x + y^2 + 8 = 0 \\ xy^2 + x - 10y + 8 = 0 \\ X_{(0)} = (0, 0)^T \end{cases}$$

- Función 2 :

$$\begin{cases} x_1^2 + x_2 - 37 = 0 \\ x_1 - x_2^2 - 5 = 0 \\ x_1 + x_2 + x_3 - 3 = 0 \\ X_{(0)} = (0, 0, 0)^T \end{cases}$$

EL cual se ocupo un error para evaluar cada una de las funciones de:

$$E = 10^{-15} \quad (43)$$

### 3.2. Funciones para sistema de ecuaciones lineales

Los sistemas de ecuaciones propuestos para el desarrollo de la segunda parte de laboratorio fueron los siguientes:

- Sistema 289.
- Sistema 1084.
- Sistema 4225.

Para el análisis de estas ecuaciones propuestas se ocuparan los siguientes métodos para sistema de ecuaciones lineales:

- LU.
- Croud.

- Doolittle.
- QR.
- Givens.
- Housholder.

## 4. Resultados

En este capítulo entregaremos cada uno de los resultados obtenidos al aplicar cada uno de los métodos en matlab.

### 4.1. Resultado de ecuaciones no lineales

Tenemos 3 funciones a evaluar las cuales son las siguientes :

- Función  $f(x) = 2x - 2^{-x}$  :

Tenemos que tener en cuenta que según como informe el Cuadro 1, los puntos escogidos para esta función son  $x_0 = 0$  y  $x_1 = 2$ , dejando excepto de estos dos puntos al método Newton-Raphson el cual solamente toma como punto a  $x_0 = 0$ .

Dejar en claro que se presentará el costo espacial juntos, lo cual en la sección de análisis de resultados tenemos los costos espaciales agrupados en dos grupos los cuales el primero consta de suma y resta y el segundo grupo conformado por multiplicación y división.

Método	Aproximación	Error	Iteraciones	Costo espacial	Tiempo
Bisección	0,641185744504986	0	56	275	0,001153
De la Secante	0,641185744504986	0	7	53	0,000254
Regular Falsi	0,641185744504986	0	17	136	0,000242
Newton-Raphson	0,641185744504986	0	6	20	0,000245

Cuadro 4: Tabla de resultados para la función (1)

A continuación se deja en evidencia los gráficos que se obtuvieron al ejecutar el programa en Matlab, los cuales no ayudan a entender el comportamiento de cada uno de los 4 métodos ocupados para esta sección del laboratorio.

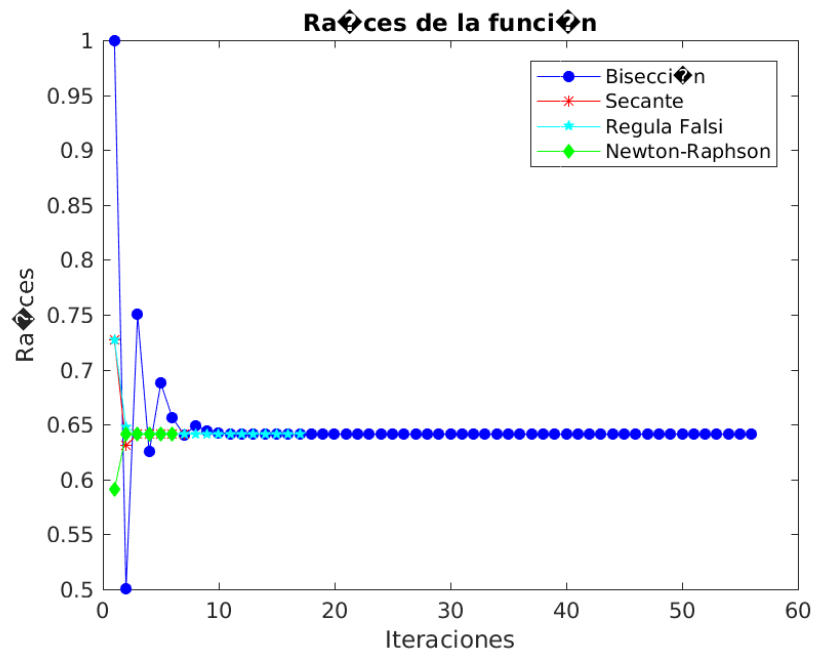


Figura 1: Resultado de las aproximaciones de la función 1

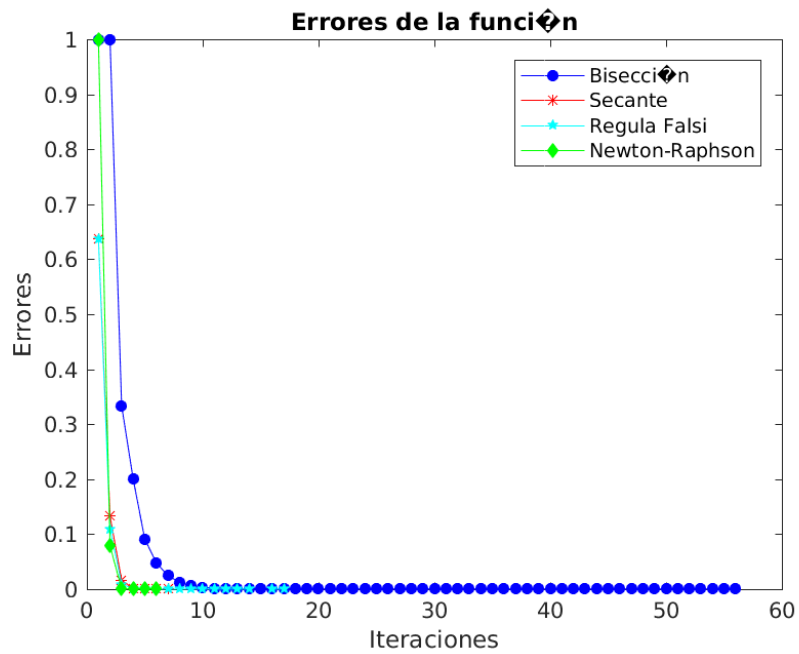


Figura 2: Resultados de los errores obtenidos en la función 1

- Función  $f(x) = e^{-x} - x^2 + 3x - 2$  :

Tenemos que tener en cuenta que según como informe el Cuadro 1, los puntos escogidos

para esta función son  $x_0 = 0$  y  $x_1 = 1$ , dejando excepto de estos dos puntos al método Newton-Raphson el cual solamente toma como punto a  $x_0 = 0$ .

Dejar en claro que se presentará el costo espacial juntos, lo cual en la sección de análisis de resultados tenemos los costos espaciales agrupados en dos grupos los cuales el primero consta de suma y resta y el segundo grupo conformado por multiplicación y división.

Método	Aproximación	Error	Iteraciones	Costo espacial	Tiempo
Bisección	0,608989103010166	0	55	270	0,001032
De la Secante	0,608989103010166	0	8	74	0,000211
Regular Falsi	0,608989103010166	0	14	224	0,000322
Newton-Raphson	0,608989103010166	0	7	24	0,000150

Cuadro 5: Tabla de resultados para la función (2)

A continuación se deja en evidencia los gráficos que se obtuvieron al ejecutar el programa en Matlab, los cuales no ayudan a entender el comportamiento de cada uno de los 4 métodos ocupados para esta sección del laboratorio.

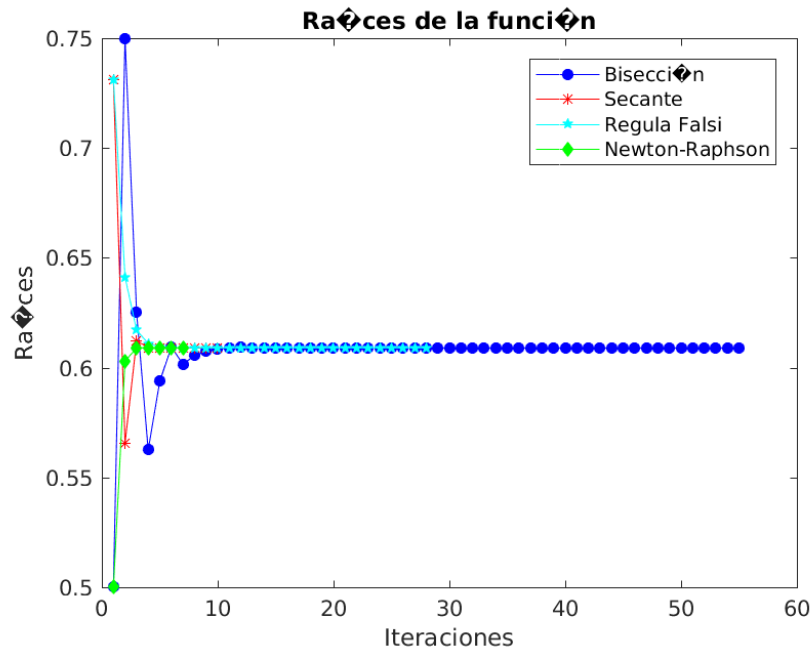


Figura 3: Resultado de las aproximaciones de la función 1

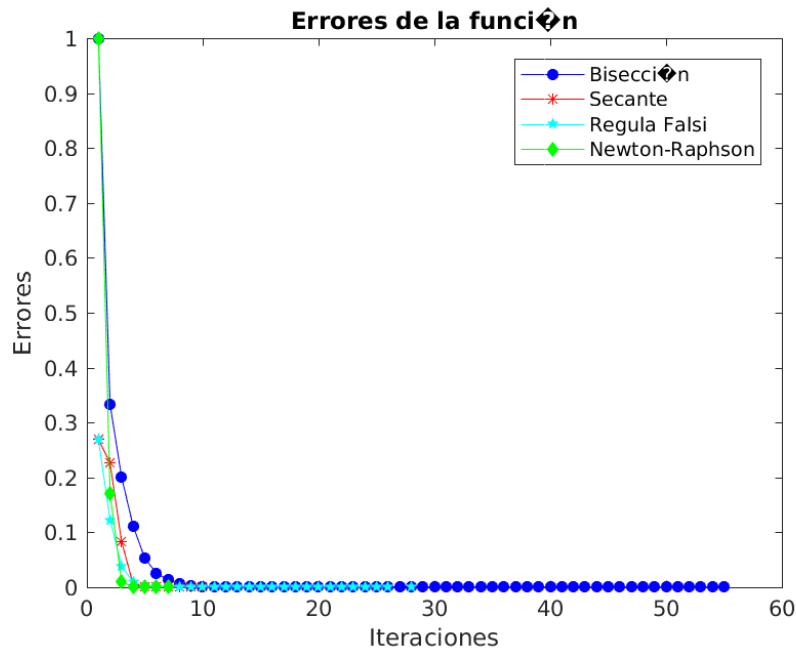


Figura 4: Resultados de los errores obtenidos en la función 1

- Función  $f(x) = x^3 - 7x^2 + 14\sin(2 * \pi * x) - 6$  :

Tenemos que tener en cuenta que según como informe el Cuadro 1, los puntos escogidos para esta función son  $x_0 = 0$  y  $x_1 = 2$ , dejando excepto de estos dos puntos al método Newton-Raphson el cual solamente toma como punto a  $x_0 = 0$ .

Dejar en claro que se presentará el costo espacial juntos, lo cual en la sección de análisis de resultados tenemos los costos espaciales agrupados en dos grupos los cuales el primero consta de suma y resta y el segundo grupo conformado por multiplicación y división.

Método	Aproximación	Error	Iteraciones	Costo espacial	Tiempo
Bisección	0,0709304012362168	0	53	265	0,002565
De la Secante	0,0709304012362168	0	8	67	0,000724
Regular Falsi	0,0709304012362168	0	16	136	0,000603
Newton-Raphson	0,0709304012362168	0	100	400	0,001598

Cuadro 6: Tabla de resultados para la función (3)

A continuación se deja en evidencia los gráficos que se obtuvieron al ejecutar el pro-

grama en Matlab, los cuales no ayudan a entender el comportamiento de cada uno de los 4 métodos ocupados para esta sección del laboratorio.

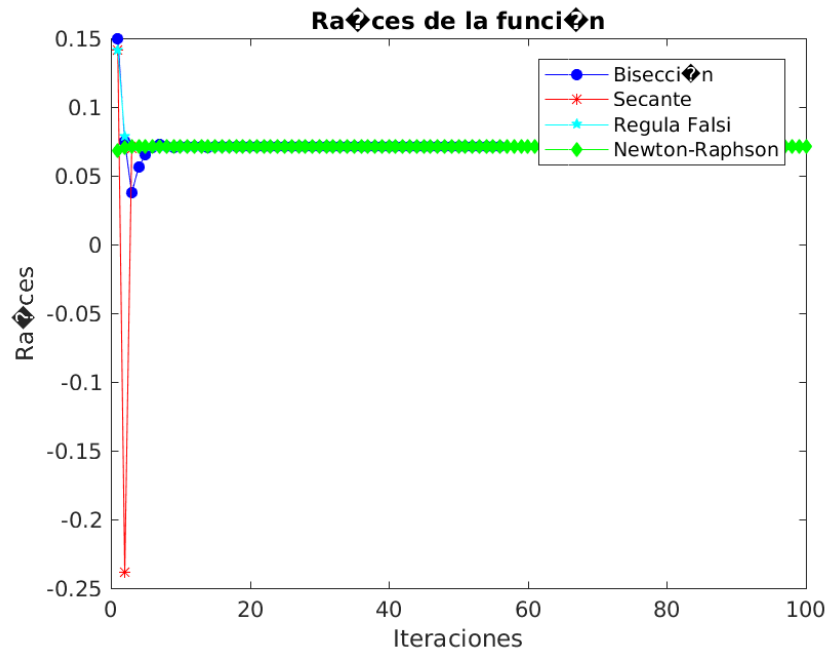


Figura 5: Resultado de las aproximaciones de la función 3

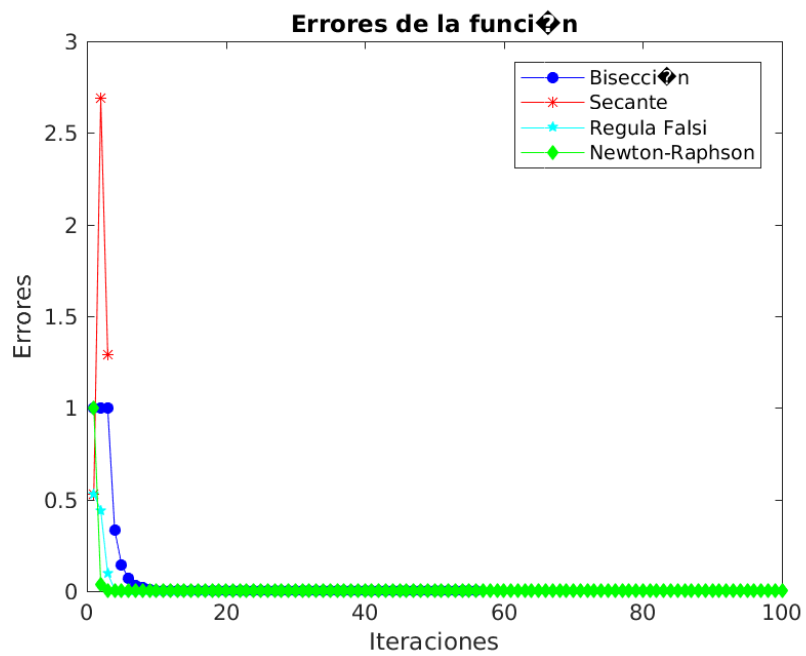


Figura 6: Resultados de los errores obtenidos en la función 3



## 4.2. Resultado de sistema de ecuaciones no lineal

Ahora presentaremos los dos sistemas de ecuaciones que trabajamos, los cuales fueron :

■ Función 1 :

$$\begin{cases} x^2 - 10x + y^2 + 8 = 0 \\ xy^2 + x - 10y + 8 = 0 \\ X_{(0)} = (0, 0)^T \end{cases}$$

Variable	Aproximación
$x_1$	1
$x_2$	1

Cuadro 7: Tabla de aproximaciones para sistema de ecuación no lineal - Función 1

Tiempo	Costos espacial
0,4354570000000000	15

Cuadro 8: Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 1

Dejar en claro que se presentará el costo espacial juntos, lo cual en la sección de análisis de resultados tenemos los costos espaciales agrupados en dos grupos los cuales el primero consta de suma y resta y el segundo grupo conformado por multiplicación y división.

Iteraciones	Error
1	1
2	0,193375370265425
3	0,00825699183787482
4	3,14752060370504e-05
5	3,93470811488328e-10
6	0

Cuadro 9: Tabla de errores obtenidos para sistema de ecuaciones no lineal - Función 1

A continuación se deja en evidencia los gráficos que se obtuvieron al ejecutar el programa en Matlab, los cuales no ayudan a entender el comportamiento que tuvo la función 1 durante el tiempo.

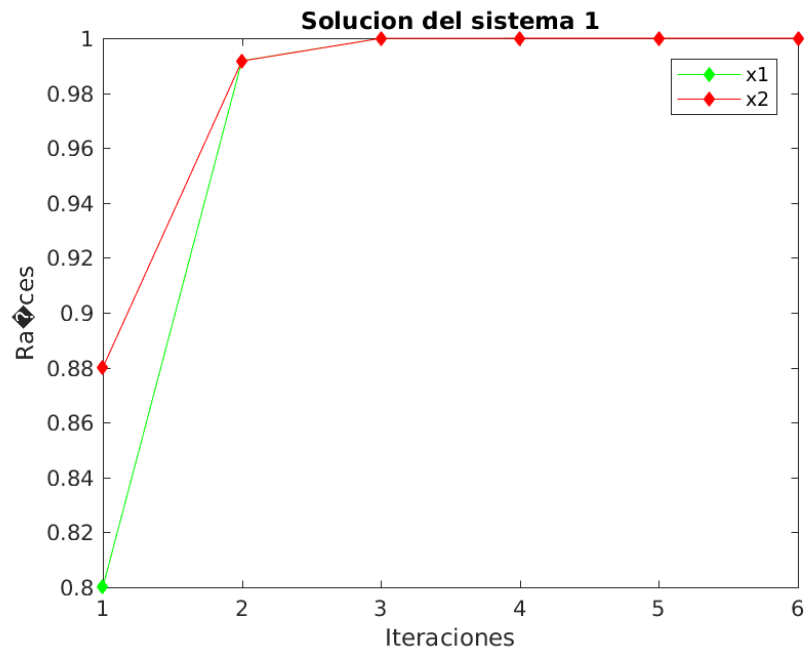


Figura 7: Resultado de las aproximaciones de la función 1 de sistema de ecuaciones no lineal

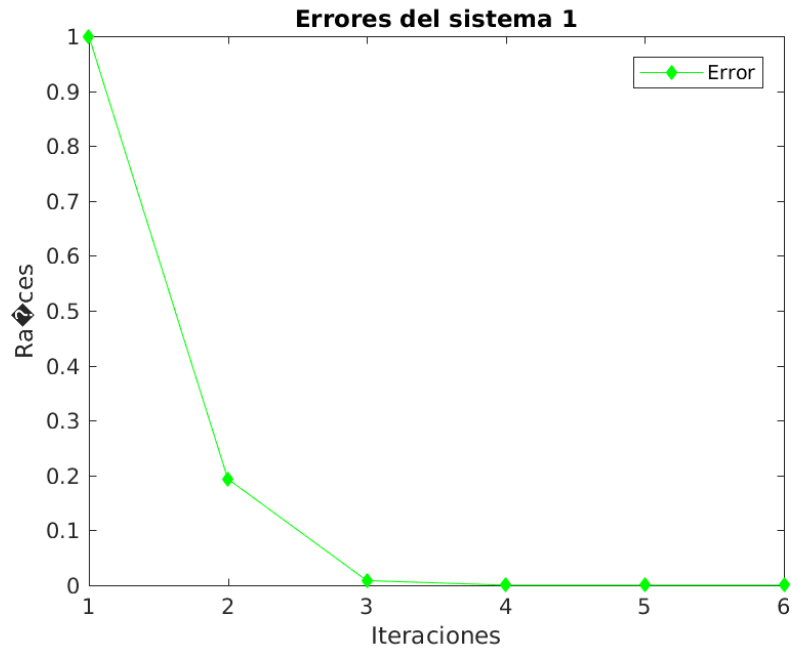


Figura 8: Resultados de los errores obtenidos en la función 1 de sistema de ecuaciones no lineal

■ Función 2 :

$$\left\{ \begin{array}{l} x_1^2 + x_2 - 37 = 0 \\ x_1 - x_2^2 - 5 = 0 \\ x_1 + x_2 + x_3 - 3 = 0 \\ X_{(0)} = (0, 0, 0)^T \end{array} \right.$$

Variable	Aproximación
$x_1$	5,99999999999995
$x_2$	1
$x_3$	-4

Cuadro 10: Tabla de aproximaciones para sistema de ecuación no lineal - Función 2

Dejar en claro que se presentará el costo espacial juntos, lo cual en la sección de análisis de resultados tenemos los costos espaciales agrupados en dos grupos los cuales el primero consta de suma y resta y el segundo grupo conformado por multiplicación y división.

<b>Tiempo</b>	<b>Costos espacial</b>
1,28484800000000	33

Cuadro 11: Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal  
- Función 2

Iteración	Error
1	1
2	0,965517241379310
3	0,794866763971107
4	0,623540825602279
5	0,380437934427999
6	0,170066355455615
7	0,0590609535488668
8	0,00950126252125771
9	0,000259735146638770
10	$1,94400632258509e^{-7}$
11	$1,08912878715729e^{-13}$
12	0

Cuadro 12: Resultados de los errores obtenidos en la función 1 de sistema de ecuaciones no lineal

A continuación se deja en evidencia los gráficos que se obtuvieron al ejecutar el programa en Matlab, los cuales no ayudan a entender el comportamiento que tuvo la función 2 durante el tiempo.

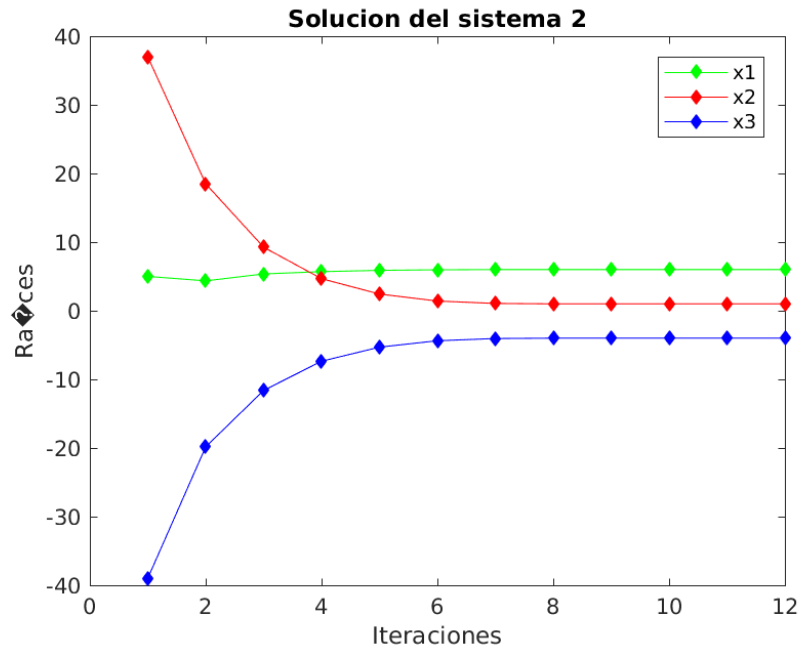


Figura 9: Resultado de las aproximaciones de la función 2 de sistema de ecuaciones no lineal

Ahora por ultimo, si queremos apreciar en formato 3D la función 2, esta nos forma algo como:

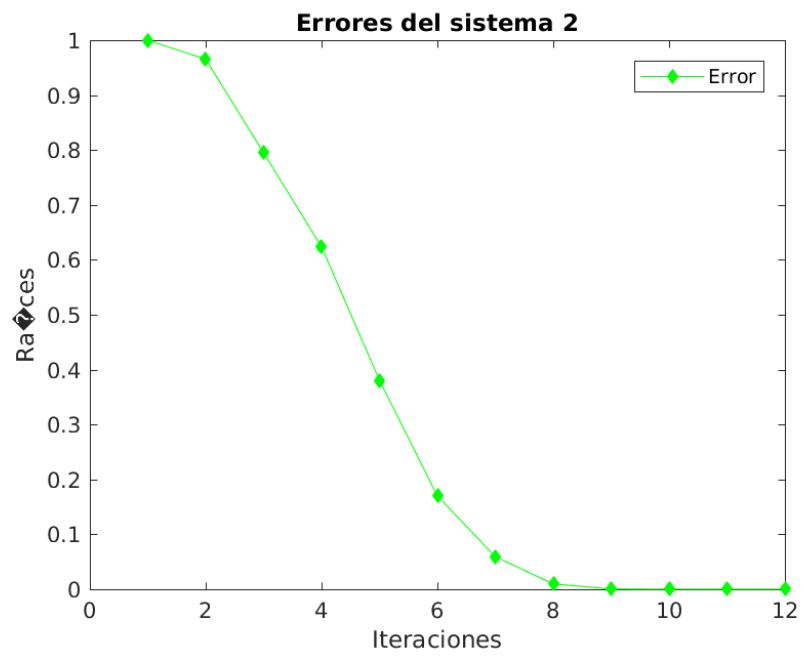


Figura 10: Resultados de los errores obtenidos en la función 2 de sistema de ecuaciones no lineal

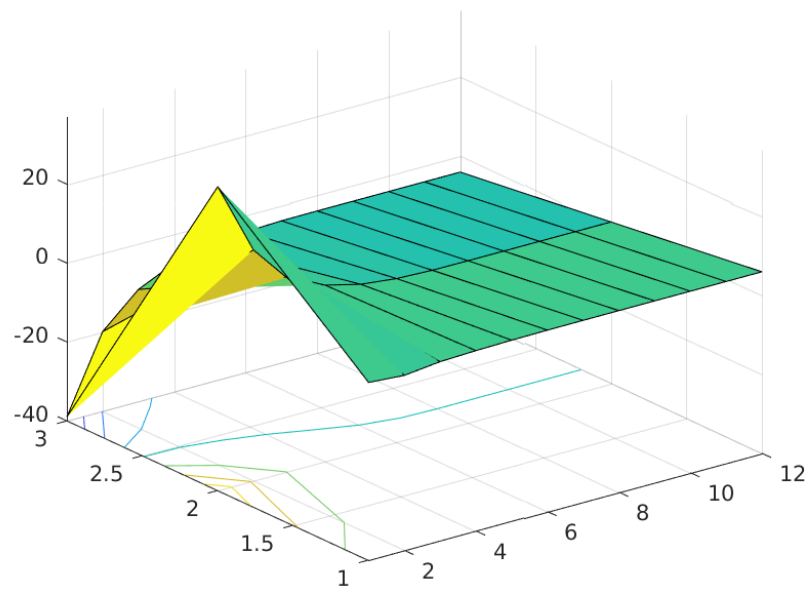


Figura 11: Apreciación en formato 3D de la función 2 de sistema de ecuaciones no lineal

### 4.3. Resultado de sistema de ecuaciones lineales

Tener en cuenta que la tolerancia utilizada para el desarrollo de cada uno de estos métodos fue de :

$$E = 10^{-18} \quad (44)$$

EL desarrollo se aplico para 3 matrices las cuales son las siguientes:

- Matriz 289

Las dimensiones de esta matriz recaen en 289 x 289, al resolver los sistemas directos e iterativos con dicha matriz, se obtiene los siguientes gráfico, los cuales nos van a representar el comportamiento que tuvo durante el lapso de tiempo:

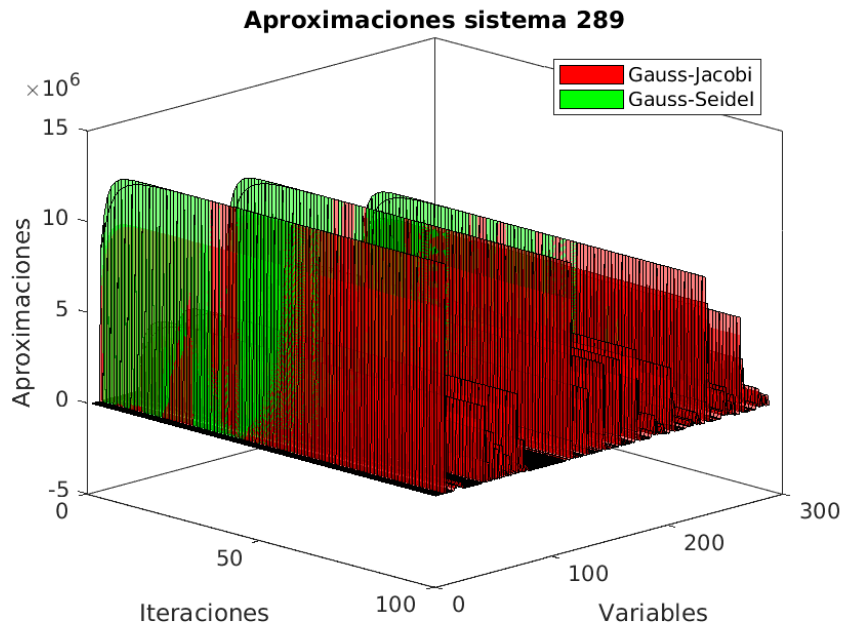


Figura 12: Aproximación de la matriz 289x289 por métodos iterativos

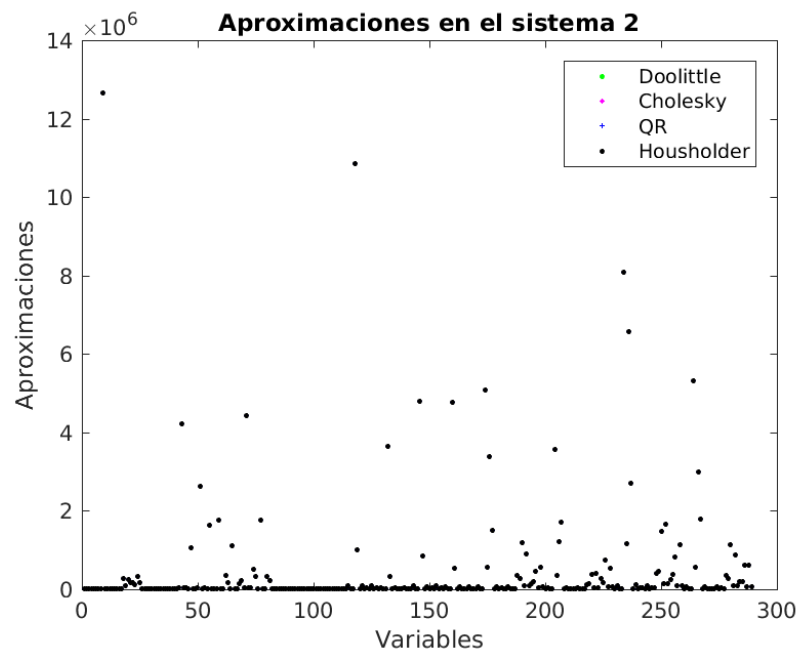


Figura 13: Aproximación de la matriz 289x289 por métodos directos

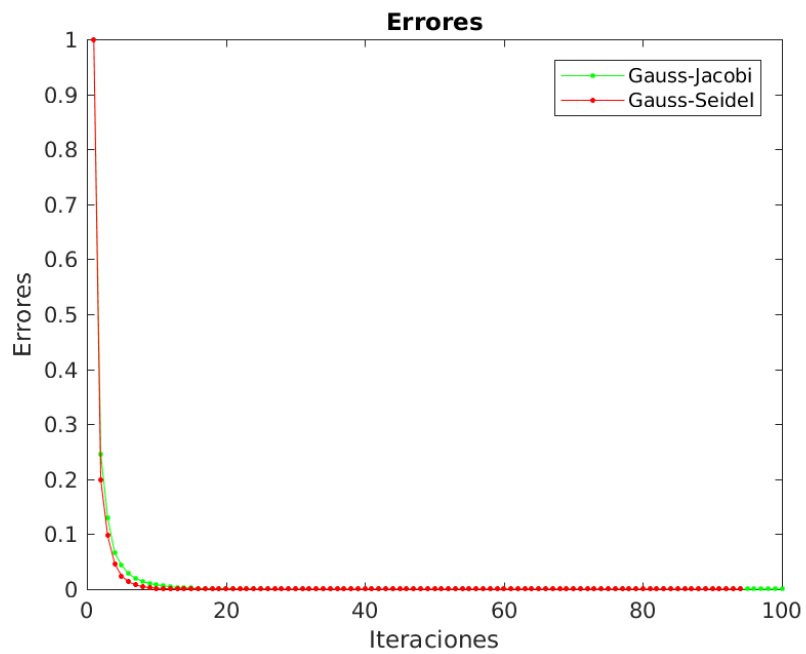


Figura 14: Error obtenido en el desarrollo de la matriz de 289x289

A modo de resumen de los 3 gráficos entregados, se construye la siguiente tabla de datos :



Al resumir la información obtenida, se tiene el siguiente cuadro:

<b>Método</b>	<b>Iteraciones</b>	<b>Error</b>	<b>Tiempo(seg)</b>
Gauss-Jacobi	100	$6,23087959808543e^{-12}$	0,7948170000000000
Gauss-Seidel	94	$2,87434439045092e^{-19}$	1,0170790000000000
Doolittle	-	$2,20615535725122e^{-15}$	0,0808400000000000
Cholesky	-	$3,15106810710579e^{-15}$	0,1763880000000000
QR	-	$2,84808638845605e^{-15}$	0,1417300000000000
Givens		$8,36125988513487e^{-15}$	0,2125170000000000
Housholder	-	$4,83538485660699e^{-15}$	0,0825170000000000

Cuadro 13: Tabla de resumen para matriz de 289x289.

A continuación se deja expresado cada uno de los costos espaciales que tuvo cada método :

<b>Método</b>	<b>Costos espaciales</b>
Gauss-Jacobi	989
Gauss-Seidel	1.041
Doolittle	16.091.525
Cholesky	83.815
QR	125.142
Givens	167.776
Housholder	3.762

Cuadro 14: Tabla de de costos espaciales para matriz de 289x289

## ■ Matriz 1089

Las dimensiones de esta matriz recaen en  $1089 \times 1089$ , al resolver los sistemas directos e iterativos con dicha matriz, se obtiene los siguientes gráfico, los cuales nos van a representar el comportamiento que tuvo durante el lapso de tiempo:

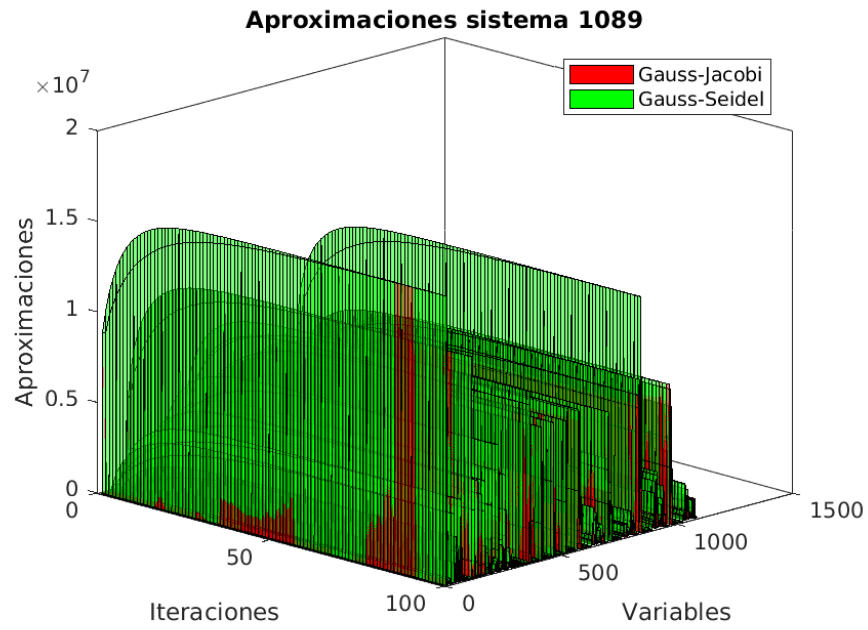


Figura 15: Aproximación de la matriz  $1089 \times 1089$  por métodos iterativos

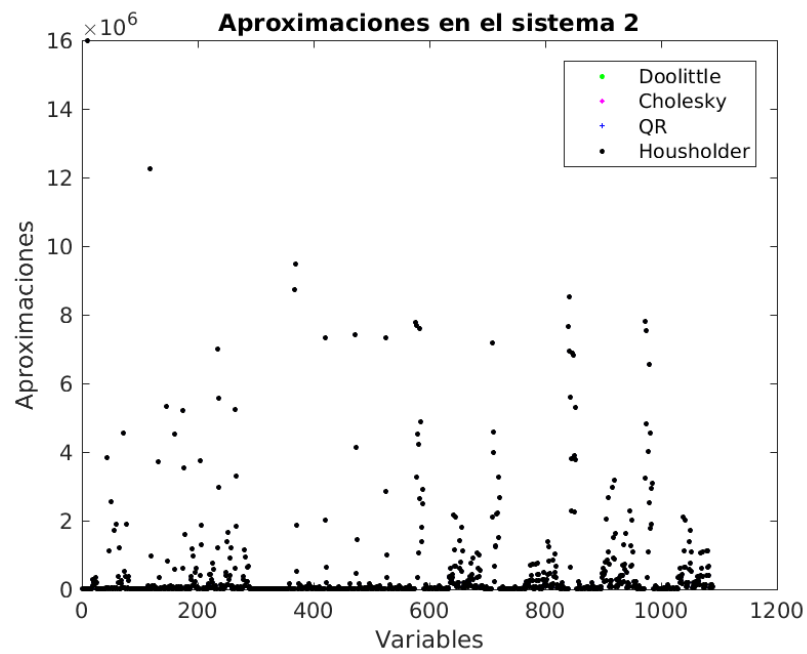


Figura 16: Aproximación de la matriz 1089x1089 por métodos directos

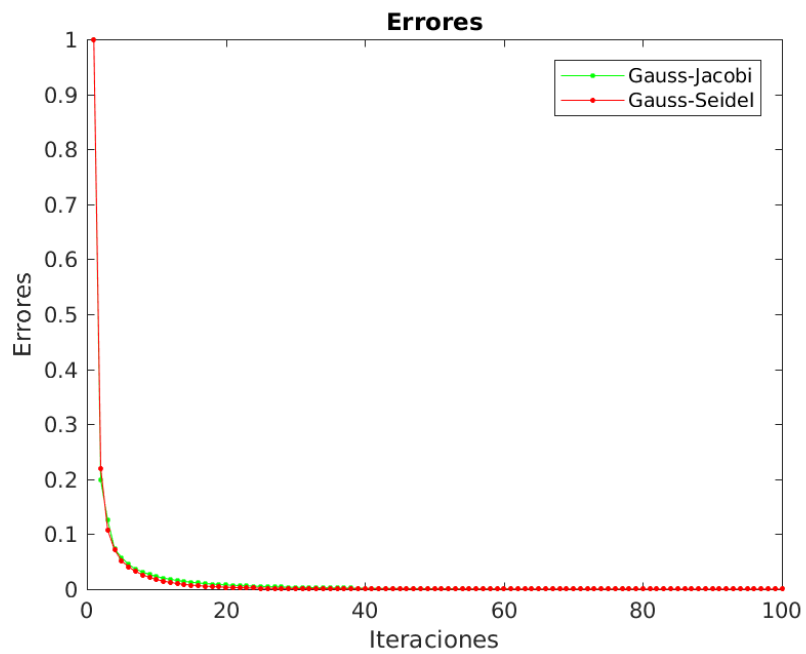


Figura 17: Error obtenido en el desarrollo de la matriz de 1089x1089

A modo de resumen de los 3 gráficos entregados, se construye la siguiente tabla de datos :

Al resumir la información obtenida, se tiene el siguiente cuadro:

<b>Método</b>	<b>Iteraciones</b>	<b>Error</b>	<b>Tiempo(seg)</b>
Gauss-Jacobi	100	$3,07771275738261e^{-05}$	11,8583120000000
Gauss-Seidel	100	$2,21062615905533e^{-07}$	17,1767500000000
Doolittle	-	$13,549748057885866e^{-14}$	6,18021400000000
Cholesky	-	$5,283192595769840e^{-14}$	4,00846700000000
QR	-	$1,180438366378053e^{-14}$	2,89086300000000
Givens	-	$5,031628960140376e^{-14}$	54,5251130000000
Housholder	-	$2,457350154221130e^{-14}$	5,02325200000000

Cuadro 15: Tabla de resumen para matriz de 1089x1089.

A continuación se deja expresado cada uno de los costos espaciales que tuvo cada método :

<b>Método</b>	<b>Costos espaciales</b>
Gauss-Jacobi	1.789
Gauss-Seidel	1.889
Doolittle	860.977.925
Cholesky	1.187.012
QR	1778342
Givens	2369502
Housholder	14.162

Cuadro 16: Tabla de de costos espaciales para matriz de 289x289

## ■ Matriz 4225

Las dimensiones de esta matriz recaen en  $4225 \times 4225$ , al resolver los sistemas directos e iterativos con dicha matriz, se obtiene los siguientes gráfico, los cuales nos van a representar el comportamiento que tuvo durante el lapso de tiempo:

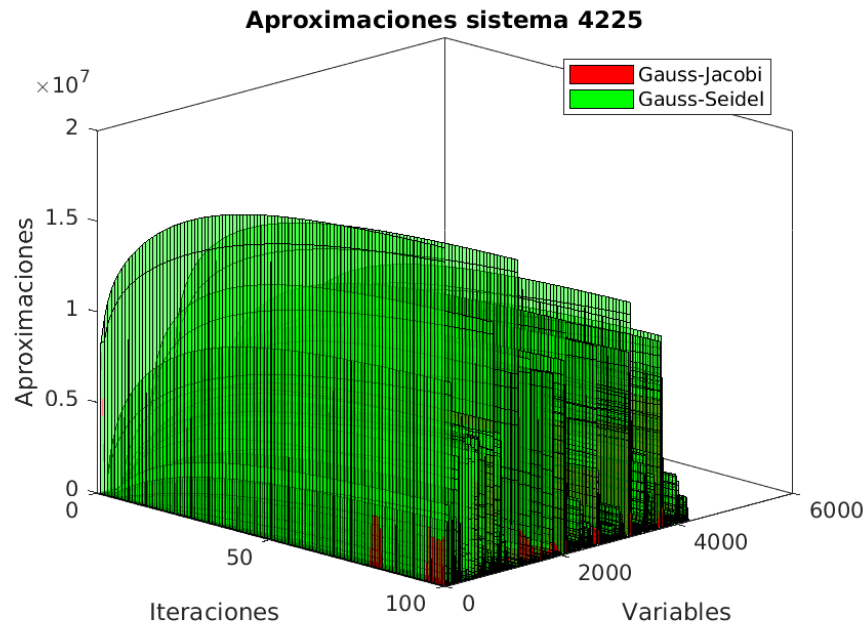


Figura 18: Aproximación de la matriz  $4225 \times 4225$  por métodos iterativos

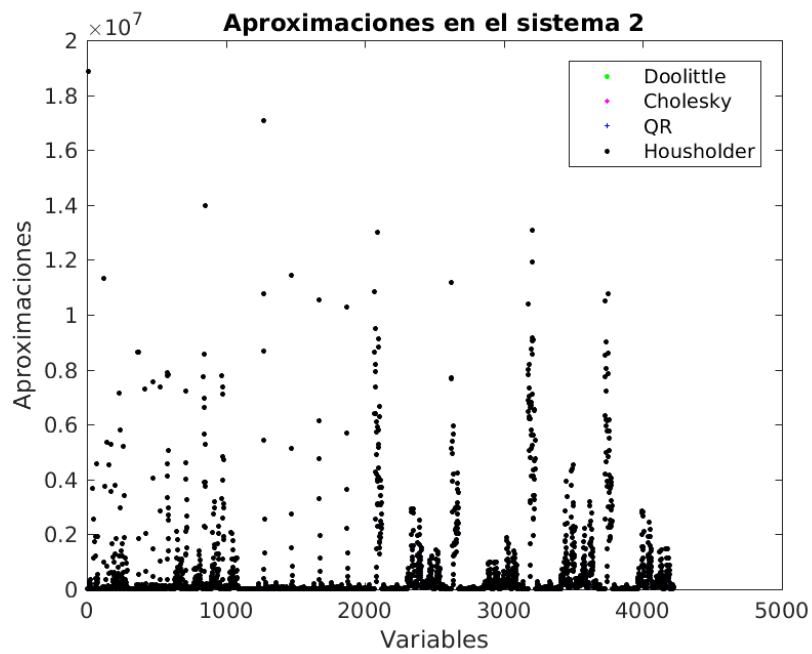


Figura 19: Aproximación de la matriz 4225x4225 por métodos directos

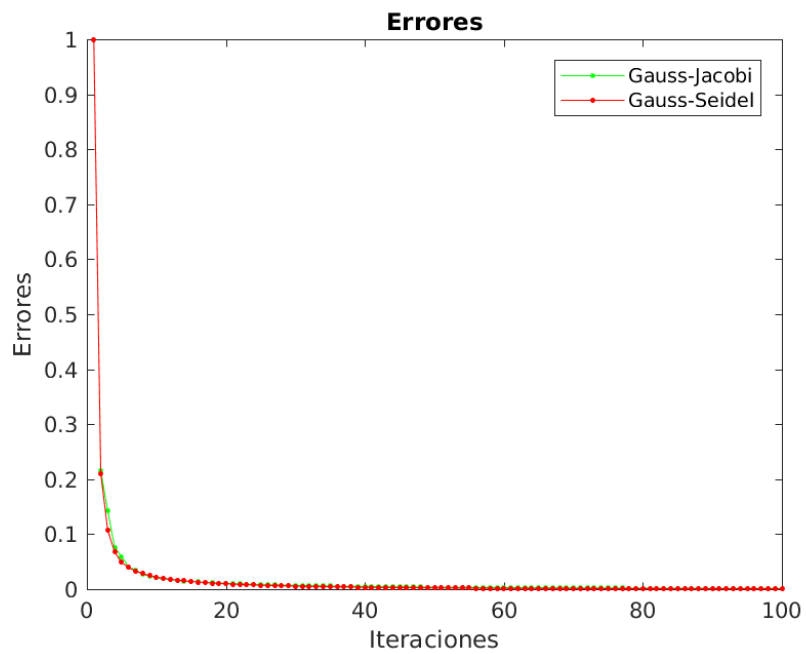


Figura 20: Error obtenido en el desarrollo de la matriz de 4225x4225

A modo de resumen de los 3 gráficos entregados, se construye la siguiente tabla de datos :

Al resumir la información obtenida, se tiene el siguiente cuadro:

Método	Iteraciones	Error	Tiempo(seg)
Gauss-Jacobi	100	0,00100843829115138	$1,1220924370000000e^3$
Gauss-Seidel	100	0,000317222444411699	$7,8640853300000000e^2$
Doolittle	-	$65,130340483177260e^{-14}$	$5,4554402000000000e^2$
Cholesky	-	$27,307890152161573e^{-14}$	$1,1595007900000000e^2$
QR	-	$8,531376288060595e^{-14}$	$3,4313634900000000e^{02}$
Givens	-	-	-
Housholder	-	$1,379119490520927e^{-13}$	$9,6805462100000000e^{02}$

Cuadro 17: Tabla de resumen para matriz 4225x4225.

A continuación se deja expresado cada uno de los costos espaciales que tuvo cada método :

Método	Costos espaciales
Gauss-Jacobi	4.925
Gauss-Seidel	5.025
Doolittle	$1,366731923e^{10}$
Cholesky	17.854.855
QR	26.773.830
Givens	-
Housholder	54.930

Cuadro 18: Tabla de de costos espaciales para matriz de 289x289

## 5. Análisis de resultados

En este capítulo tendremos el análisis de los comportamientos de cada uno de los métodos usados para cada una de las funciones. Por lo cual, se analizará cada uno de los gráficos y tablas que fueron entregados en el Capítulo de Resultados.

### 5.1. Métodos no lineales

#### 5.1.1. Función 1 : $f(x) = 2x - 2^{-x}$

Lo primero que tenemos que dejar en claro que esta función solamente consta de una raíz, la cual se produce cuando  $x = 2$ . Por ende, no tenemos complicaciones de saber cual es la raíz que se está analizando. Además, se debe saber que para el método de Newton-Raphson se utiliza el criterio de la regla de Fourier para escoger el punto inicial, el cual en esta ocasión recae en  $x_0 = 2$ .

Ahora procedemos a ordenar los métodos según su número de iteraciones quedando de la siguiente forma :

Método	Iteraciones
Bisección	56
Regular Falsi	17
Secante	7
Newton-Raphson	6

Cuadro 19: Tabla de mayor iteración

Esto nos indica que el método que menos se acerca a la raíz es Bisección y el método que más se acerca a la raíz es Newton-Raphson.

Al interpretar el gráfico de errores tenemos que el que se demora más en acercarse al eje de las abscisas es:

- Bisección
- Secante



- Regular Falsi
- Newton-Raphson

Cabe de tacar un detalle que el método Regular Falsi y Secante tiene una diferencia muy menor al momento de alcanzar el eje de las abscisas.

A continuación se realizara un ranking según el costo espacial que produce cada método implementado:

<b>Método</b>	<b>Costo espacial (+ y -)</b>	<b>Costo espacial (/ y *)</b>
Newton-Raphson	10	10
Secante	30	23
Regular Falsi	68	68
Bisección	110	165

Cuadro 20: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor costo espacial es Bisección y por el otro extremo el que produce un menor costo espacial es el método de Newton-Raphson.

Por ultimo, se deja un ranking acerca de los tiempos de ejecución que realizan cada uno de los métodos implementados:

<b>Método</b>	<b>Tiempo</b>
Regular Falis	0.000242
Newton-Raphson	0.000245
Secante	0.000254
Bisección	0.001153

Cuadro 21: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor tiempo de ejecución es Bisección y por el otro extremo el que produce un menor tiempo de ejecución es el método de Regular Falsi.

### 5.1.2. Función 2 : $f(x) = e^{-x} - x^2 + 3x - 2$

Lo primero que tenemos que dejar en claro que esta función consta con mas de una raíz, aunque tomaremos un intervalo definida por  $[0, 1]$ , el cual consta con una raíz, por ende, sabremos que raíz se esta calculando con cada uno de los métodos. Además, se debe saber que para el método de Newton-Raphson se utiliza el criterio de la regla de fourier para escoger el punto inicial, el cual en esta ocasión recae en  $x_0 = 0$ .

Ahora procedemos a ordenar los métodos según su numero de iteraciones quedando de la siguiente forma :

Método	Iteraciones
Bisección	55
Regular Falsi	14
Secante	8
Newton-Raphson	7

Cuadro 22: Tabla de mayor iteración

Esto nos indica que el método que menos se acerca a la raíz es Bisección y el método que mas se acerca a la raíz en Newton-Raphso.

Al interpretar el gráfico de errores tenemos que el se demora mas en acerca al eje de las abscisas es:

- Bisección
- Secante
- Regular Falsi
- Newton-Raphson

Cabe de tacar un detalle que el método Regular Falsi y Secante tiene una diferencia muy menor al momento de alcanzar el eje de las abscisas. Además, destacar que el comportamiento que tiene Newton-Raphson y Regular Falsi se asemejan en cierto rigor al momento que se acercan al eje de las abscisas, dejar en claro que esto no significa que ambos sean

igual de eficiente solamente que comparten una cierto acercamiento en el comportamiento que obtienen en el tiempo.

A continuación se realizara un ranking según el costo espacial que produce cada método implementado:

<b>Método</b>	<b>Costo espacial (+ y -)</b>	<b>Costo espacial (/ y *)</b>
Newton-Raphson	12	12
Secante	42	32
Regular Falsi	112	112
Bisección	108	162

Cuadro 23: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor costo espacial es Bisección y por el otro extremo el que produce un menor costo espacial es el método de Newton-Raphson.

Por ultimo, se deja un ranking acerca de los tiempos de ejecución que realizan cada uno de los métodos implementados:

<b>Método</b>	<b>Tiempo</b>
Newton-Raphson	0.000150
Secante	0.000211
Regular Falsi	0.000322
Bisección	0.001032

Cuadro 24: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor tiempo de ejecución es Bisección y por el otro extremo el que produce un menor tiempo de ejecución es el método de Newton-Raphson.

### 5.1.3. Función 3 : $f(x) = x^3 - 7x^2 + 14\sin(2 * \pi * x) - 6$

Lo primero que tenemos que dejar en claro que esta función consta con mas de una raíz, aunque tomaremos un intervalo definida por  $[0, 1]$ , el cual consta con una raíz, por ende, sabremos que raíz se esta calculando con cada uno de los métodos. Además, se debe saber que para el método de Newton-Raphson se utiliza el criterio de la regla de fourier para escoger el punto inicial, el cual en esta ocasión recae en  $x_0 = 0$ .

Ahora procedemos a ordenar los métodos según su numero de iteraciones quedando de la siguiente forma :

Método	Iteraciones
Bisección	55
Regular Falsi	14
Secante	8
Newton-Raphson	7

Cuadro 25: Tabla de mayor iteración

Esto nos indica que el método que menos se acerca a la raíz es Bisección y el método que mas se acerca a la raíz en Newton-Raphso.

Al interpretar el gráfico de errores tenemos que el se demora mas en acerca al eje de las abscisas es:

- Bisección
- Secante
- Regular Falsi
- Newton-Raphson

Cabe de tacar un detalle que el método Regular Falsi y Secante tiene una diferencia muy menor al momento de alcanzar el eje de las abscisas.

A continuación se realizara un ranking según el costo espacial que produce cada método implementado:

<b>Método</b>	<b>Costo espacial (+ y -)</b>	<b>Costo espacial (/ y *)</b>
Secante	38	29
Regular Falsi	68	68
Bisección	106	159
Newton-Raphson	200	200

Cuadro 26: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor costo espacial es Newton-Raphson y por el otro extremo el que produce un menor costo espacial es el método de Secante.

Por ultimo, se deja un ranking acerca de los tiempos de ejecución que realizan cada uno de los métodos implementados:

<b>Método</b>	<b>Tiempo</b>
Secante	0.000724
Regular Falsi	0.000603
Bisección	0.002565
Newton-Raphson	0.001598

Cuadro 27: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor tiempo de ejecución es Newton-Raphson y por el otro extremo el que produce un menor tiempo de ejecución es el método de Secante.

Se deja en claro que a diferencia de las dos instrucciones anteriores el método de Newton-Raphson había tenido un comportamiento donde lo calificaba como uno de los métodos, acá donde la función tiene un nivel mayor en su derivación, perjudica ha dicho método posicionándolo como el peor método dentro de los que se esta analizando.

## 5.2. Sistema de ecuaciones no lineales

### 5.2.1. Sistema de ecuaciones 1 :

Para empezar se puede dejar en claro que los valores obtenidos son muy cercanos a los reales, es decir, que podemos asegurar que los valores obtenidos son exitosos y cumplirían con el sistema de ecuaciones.

Ahora por otro lado, analizando la tabla de errores y su respectivo gráfico, se puede corroborar que los valores obtenidos están en lo correcto, ya que su error finalmente se expreso como  $E = 0$ .

El análisis de la siguiente tabla consiste en :

Tiempo	Costos espacial (+ y -)	Costos espacial (/ y *)
0,435457000000000	10	5

Cuadro 28: Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 1

Se puede afirmar que el metodo tiene un tiempo de ejecución que recae en aproximadamente 0,4354570. Además, tenemos que su costo espacial que tiene recae en  $C_{espacial_{total}} = 15$ , repartidos por cada categoría de la siguiente forma  $C_{espacial_{(+y-)}} = 10$  y  $C_{espacial_{(*y/)}} = 5$ , se deja en claro que el costo espacial no considera ciertos cálculos en este método, por el motivo que se ocupan funciones propias de 'Matlab' las cuales no son consideradas dentro del calculo de operaciones básicas.

Por ultimo, se tiene que para cumplir con la tolerancia con la que fueron probados los sistema de ecuaciones, basto con realizar 6 iteraciones para poder lograr el objetivo y entregar con un error mínimo los valores de  $x_1$  y  $x_2$ .

### 5.2.2. Sistema de ecuaciones 2 :

Para empezar se puede dejar en claro que los valores obtenidos son muy cercanos a los reales, es decir, que podemos asegurar que los valores obtenidos son exitosos y cumplirían con el sistema de ecuaciones.

Ahora por otro lado, analizando la tabla de errores y su respectivo gráfico, se

puede corroborar que los valores obtenidos están en lo correcto, ya que su error finalmente se expreso como  $E = 0$ .

El análisis de la siguiente tabla consiste en :

Tiempo	Costos espacial (+ y -)	Costos espacial (/ y *)
1,28484800000000	22	11

Cuadro 29: Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 2

Se puede afirmar que el metodo tiene un tiempo de ejecución que recae en aproximadamente 1,2848480. Además, tenemos que su costo espacial que tiene recae en  $C_{espacial_{total}} = 33$ , repartidos por cada categoría de la siguiente forma  $C_{espacial_{(+y-)}} = 22$  y  $C_{espacial_{(*y/)}} = 11$ , se deja en claro que el costo espacial no considera ciertos cálculos en este método, por el motivo que se ocupan funciones propias de 'Matlab' las cuales no son consideradas dentro del calculo de operaciones básicas.

Por ultimo, se tiene que para cumplir con la tolerancia con la que fueron probados los sistema de ecuaciones, basto con realizar 11 iteraciones para poder lograr el objetivo y entregar con un error mínimo los valores de  $x_1$  ,  $x_2$  y  $x_3$ .

A modo de comparación entre ambos sistemas de ecuaciones tanto como para le sistema de ecuaciones que consta con 2 variables y el sistema que consta con 3 variables, ambos lograron llegar a resultados exitosos, obteniendo en ambas oportunidad un error  $E = 0$ , por ende, se tiene que destacar que para poder cumplir dicha exigencia el sistema de ecuaciones con 3 variables necesito de 5 iteraciones extras que el sistema de ecuaciones con 2 variables.

## 5.3. Métodos lineales

### 5.3.1. Matriz 289x289 :

Procederemos analizar cada uno de lo gráficos entregados en el capitulo anterior.

Por una parte tenemos que en la figura 12 nos indica que la convergencia de los métodos iterativos. Además, si nos dirigimos a revisar el Cuadro 11, tenemos que el método Gauss-Jacobi realizo un total de 6 iteraciones mas que el método Gauus-Sield.

Por otra parte tenemos que en la figura 13 se nos indica que la convergencia que se produce en la misma independiente el método que se este ocupando.

Por ultimo, se procede a entregar una tabla con los tiempos mayores de ejecución de cada uno de los métodos de sistema de ecuaciones lineales.

<b>Método</b>	<b>Tiempo(seg)</b>
Doolittle	0,0808400000000000
Housholder	0,0825170000000000
QR	0,1417300000000000
Cholesky	0,1763880000000000
Gauss-Jacobi	0,7948170000000000
Gauss-Seidel	1,0170790000000000
Givens	0,2125170000000000

Cuadro 30: Tabla de resumen tiempo de ejecución para sistema 289x289.

A continuación hacemos el analisis de los costos espaciales que obtuvimos:

<b>Método</b>	<b>Costos espaciales (+ y -)</b>	<b>Costos espaciales (/ y *)</b>
Gauss-Jacobi	589	400
Gauss-Seidel	665	376
Doolittle	8045761	8045764
Cholesky	41906	41909
QR	41617	83525
Givens	24216	143560
Housholder	868	2894

Cuadro 31: Tabla de de costos espaciales para matriz de 289x289

Donde nos podemos dar cuenta que los métodos iterativos tiene considerablemente un mejor rendimiento mirado desde el punto de vista de los costos espaciales que realizan. Además, se destaca que 'Gauss-Sield' tiene un mejor rendimiento que 'Gauss-Jaobi', aunque tenga un costo espacial (+ y -) menor, ya que la diferencia se hace en el costo espacial (\* y



/) el cual tiene un mayor valor agregado, por ende, es mucho mas costoso tener mas costo espacial (\* y /) que costos espacial (\* y -).

Además, podemos afirmar que el método que mas costo espacial tiene tanto como los costos espaciales (+ y - ) y costo espacial (\* y /), es el método de Doolitte, superando por mucho a los demás métodos expuestos en el cuadro 33, tanto como es su costo espacial (+ y -) y costo espacial (\* y /). Además, tenemos que el método Housholder tiene un menor costo espacial tanto como costo espacial (\* y /) y costo espacial (+ y -), lo cual lo califica como el mejor método directo evaluándolo desde el punto de vista del costo espacial.

### 5.3.2. Matriz 1089x1089 :

Procederemos analizar cada uno de los gráficos entregados en el capítulo anterior.

Por una parte tenemos que en la figura 15 nos indica que la convergencia de los métodos iterativos. Además, en este caso tenemos que el número de iteraciones es el mismo, lo que implica que el número de iteraciones en esta oportunidad pasa a ser un dato no importante para el análisis.

Por otra parte tenemos que en la figura 16 se nos indica que la convergencia que se produce en la misma independiente del método que se este ocupando.

Por último, se procede a entregar una tabla con los tiempos mayores de ejecución de cada uno de los métodos de sistema de ecuaciones lineales.

Método	Tiempo(seg)
QR	2,890863000000000
Cholesky	4,008467000000000
Housholder	5,023252000000000
Doolittle	6,180214000000000
Gauss-Jacobi	11,858312000000000
Gauss-Seidel	17,176750000000000
Givens	54,525113000000000

Cuadro 32: Tabla de resumen tiempo de ejecución para sistema 1089x1089.

Donde nos podemos dar cuenta que los métodos iterativos tiene considerablemente un mejor rendimiento mirado desde el punto de vista de los costos espaciales que realizan.

Además, podemos afirmar que el método que mas costo espacial tiene tanto como los costos espaciales (+ y -) y costo espacial (\* y /), es el método de Doolittle, superando por mucho a los demás métodos expuestos en el cuadro 33.

A continuación hacemos el análisis de los costos espaciales que obtuvimos:

Donde nos podemos dar cuenta que los métodos iterativos tiene considerablemente un mejor rendimiento mirado desde el punto de vista de los costos espaciales que realizan. Además, se destaca que 'Gauss-Jacobi' tiene un mejor rendimiento que 'Gauss-Jacobi', ya

Método	Costos espaciales (+ y -)	Costos espaciales (/ y *)
Gauss-Jacobi	1389	400
Gauss-Seidel	1489	400
Doolittle	430488961	430488964
Cholesky	593506	593506
QR	592417	1185925
Givens	339434	2030068
Housholder	3268	10894

Cuadro 33: Tabla de de costos espaciales para matriz de 1089x1089

que por una parte el costo espacial (\* y /) es el mismo, lo cual el análisis se centra en el costo espacial (+ y -), en donde por una diferencia muy menor queda mejor calificado el método 'Gauss Jacobi'.

Además, podemos afirmar que el método que mas costo espacial tiene tanto como los costos espaciales (+ y - ) y costo espacial (\* y /), es el método de Doolittle, superando por mucho a los demás métodos expuestos en el cuadro 31, tanto como es su costo espacial (+ y -) y costo espacial (\* y /). Además, tenemos que el método Housholder tiene un menor costo espacial tanto como costo espacial (\* y /) y costo espacial (+ y -), lo cual lo califica como el mejor método directo evaluándolo desde el punto de vista del costo espacial.

### 5.3.3. Matriz 4225x4255 :

Procederemos analizar cada uno de los gráficos entregados en el capítulo anterior.

Por una parte tenemos que en la figura 18 nos indica que la convergencia de los métodos iterativos. Además, en este caso tenemos que el número de iteraciones es el mismo, lo que implica que el número de iteraciones en esta oportunidad pasa a ser un dato no importante para el análisis.

Por otra parte tenemos que en la figura 19 se nos indica que la convergencia que se produce en la misma independiente del método que se este ocupando.

Por último, se procede a entregar una tabla con los tiempos mayores de ejecución de cada uno de los métodos de sistema de ecuaciones lineales.

Método	Tiempo(seg)
Gauss-Jacobi	$1,1220924370000000e^3$
Gauss-Seidel	$7,8640853300000000e^2$
Doolittle	$5,4554402000000000e^2$
Cholesky	$1,1595007900000000e^2$
QR	$3,4313634900000000e^{02}$
Givens	-
Housholder	$9,6805462100000000e^{02}$

Cuadro 34: Tabla de resumen tiempo de ejecución para sistema 1089x1089.

A continuación hacemos el análisis de los costos espaciales que obtuvimos:

Método	Costos espaciales (+ y -)	Costos espaciales (/ y *)
Gauss-Jacobi	4525	400
Gauss-Seidel	4625	400
Doolittle	$2,513962880100000e^{10}$	$2,513962880400000e^{10}$
Cholesky	8927426	8927429
QR	8923201	17850629
Givens	-	-
Housholder	12676	42254

Cuadro 35: Tabla de de costos espaciales para matriz de 4225x4225

Donde nos podemos dar cuenta que los métodos iterativos tiene considerablemente un mejor rendimiento mirado desde el punto de vista de los costos espaciales que realizan. Además, se destaca que 'Gauss-Jacobi' tiene un mejor rendimiento que 'Gauss-Jacobi', ya que por una parte el costo espacial (\* y /) es el mismo, lo cual el análisis se centra en el costo espacial (+ y -), en donde por una diferencia muy menor queda mejor calificado el método 'Gauss Jacobi'.

Además, podemos afirmar que el método que mas costo espacial tiene tanto como los costos espaciales (+ y - ) y costo espacial (\* y /), es el método de Doolittle, superando por mucho a los demás métodos expuestos en el cuadro 35, tanto como es su costo espacial (+ y -) y costo espacial (\* y /). Además, tenemos que el método Housholder tiene un menor costo espacial tanto como costo espacial (\* y /) y costo espacial (+ y -), lo cual lo califica como el mejor método directo evaluándolo desde el punto de vista del costo espacial.

## 6. Conclusiones

Como se pudo evidenciar en el capítulo número 5 de análisis de resultados, los métodos entregaron resultados muy finos y fueron de gran utilidad para poder encontrar las raíces de las funciones, las cuales no son sencillas de calcular. A pesar de que todos los métodos cumplen con el objetivo constan con una efectividad distintas debido a su manera de atacar las funciones.

Por otro lado tenemos los métodos no lineales, en este caso tenemos que la función que queremos aproximar influye de manera considerable en los resultados que entregan cada uno de los métodos. Debido a que la derivada de la función sea compleja lo cual puede recaer en escoger un método que no tenga que derivar para encontrar su raíz, además no tenemos que dejar de lado que los métodos pueden arrojar resultados distintos cuando tenemos alguna función que conste con múltiples raíces, ya que los métodos no necesariamente se estarán acercando a la misma raíz.

Tenemos que es de suma importancia escoger un intervalo preciso para cada método, ya que alejar el intervalo de la raíz de la función provoca que los métodos sean menos eficientes y precisos, ya que requieren de un mayor trabajo iterativo para llegar a la raíz de la función. Tener en cuenta que algunos métodos no entregaran resultados satisfactorios si el intervalo inicial no es el indicado. Una forma correcta de escoger este intervalo inicial es graficando la función previamente y además para las funciones que necesitan de un solo punto inicial se ocupa la 'Regla de Fourier' que entrega un punto preciso para el comienzo del método.

La tolerancia que se exige para cada método es un factor importante al momento de escogerlo, ya que mientras menor sea la tolerancia que se exige requiere un trabajo mayor iterativamente.

Se deja en claro que los comentarios mencionados anteriormente cumplen también para los sistemas de ecuaciones lineales que se implementaron dentro del laboratorio.

Uno de los factores externos que influyen en los resultados que se obtiene en cada uno de los métodos, es el lenguaje de programación. En este caso, recurrimos a 'Matlab' el cual usa una cantidad considerable de dígitos, sin embargo, solo recurre a mostrar una cantidad

de 16 dígitos en sus resultados. Debido a esto podemos plantear que cuando los métodos entregan un error igual a '0', tenemos que tuvo un trabajo arduo y el valor numérico del error es tan pequeño que solo lo aproxima a '0'.

El costo operacional que se considero se dividió en dos grandes grupos el primero estaba compuesto por las operaciones básicas suma y resta y el segundo grupo compuesto por las operaciones básicas de multiplicación y división. Sin embargo, deja de ser representativo al momento de ocuparlos en los métodos donde ocupamos funciones nativas de 'Matlab', ya que perdemos de contemplar lo que se procesa en esas operaciones. Además, se deja en claro que el lenguaje de programación ocupado provee funciones con una riqueza enorme para poder realizar cálculos matemáticos las cuales ayudaron en el transcurso del laboratorio.

A pesar de no poder implementar el cálculo del método 'Given' en la matriz  $4225 \times 4225$  por motivos de tiempo, ya que el computador se demoro un tiempo bastante considerable en procesarlas se dejaron por esta vez fuera de análisis, sin embargo no se descarta que para otra oportunidad si sean parte del análisis de datos. También mencionar que se cumplió con el objetivo principal del laboratorio que era crear los algoritmos y analizar cada uno de sus resultados, lo cual al terminar con éxito este laboratorio nos entrego herramientas y mayor entendimiento de cómo implementar una abstracción matemática a un computador.

# Bibliografía

Anonimo (2002-2003). Apuntes de métodos numéricos 2 o e.t.s.i. telecomunicación. [Online]  
<http://www.telecos-malaga.com/descargas/apuntes/2Curso/MN/MN-Apuntes.pdf>.

Cobos (2002). Apuntes de cálculo numérico.