



Informe de Algoritmos numéricos: Métodos Numéricos

Integrantes: Alberto Rodríguez
Curso: Algoritmos numéricos
Sección 0-L-2
Profesor(a): Óscar Rojas Díaz

5 de Septiembre de 2019

Tabla de contenidos

1. Introducción	1
1.1. Objetivos	1
1.1.1. Objetivos específicos	1
1.2. Herramientas	2
1.3. Estructura del informe	2
2. Descripción de los métodos numéricos	3
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	7
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 Doolittle	10
2.3.1.3. Método de Cholesky	12
2.3.1.4. Factorización de QR	13
2.3.2. Métodos indirectos	13
2.3.2.1. Método de Jacobi	14
2.3.2.2. Método de Gauss-Seidel	15
3. Funciones	16
3.1. Funciones para ecuaciones no lineales	16
3.2. Funciones para sistema de ecuaciones lineales	18

4. Resultados	20
4.1. Resultado de ecuaciones no lineales	20
4.2. Resultado de sistema de ecuaciones no lineal	28
4.3. Resultado de sistema de ecuaciones lineales	30
5. Análisis de resultados	39
5.1. Métodos no lineales	39
5.1.1. Función 1 : $f(x) = 2x - 2^{-x}$	39
5.1.2. Función 2 : $f(x) = x^3 - 7x^2 + 14x - 2$	41
5.2. Sistema de ecuaciones no lineales	43
5.2.1. Sistema de ecuaciones 1 :	43
5.2.2. Sistema de ecuaciones 2 :	44
5.3. Métodos lineales	45
5.3.1. Matriz 289x289 :	45
5.3.2. Matriz 1089x1089 :	48
5.3.3. Matriz 4225x4255 :	50
6. Conclusiones	52
7. Anexo	54
7.1. Métodos Numéricos	54
7.2. Objetivos no cumplidos	54
8. Bibliografía	54

Índice de figuras

1.	Resultado de las aproximaciones de la función 1	21
2.	Resultados de los errores obtenidos en la función 1	22
3.	Resultado de las aproximaciones de la función 1 con método del punto fijo .	22
4.	Resultados de los errores obtenidos en la función 1 con método del punto fijo	23
5.	Resultado de las aproximaciones de la función 2	25
6.	Resultados de los errores obtenidos en la función 2	26
7.	Resultado de las aproximaciones de la función 2 con método del punto fijo .	26
8.	Resultados de los errores obtenidos en la función 2 con método del punto fijo	27
9.	Aproximación de la matriz 289x289 por métodos iterativos	30
10.	Aproximación de la matriz 289x289 por métodos directos	31
11.	Error obtenido en el desarrollo de la matriz de 289x289	31
12.	Aproximación de la matriz 1089x1089 por métodos iterativos	33
13.	Aproximación de la matriz 1089x1089 por métodos directos	34
14.	Error obtenido en el desarrollo de la matriz de 1089x1089	34
15.	Aproximación de la matriz 4225x4225 por métodos iterativos	36
16.	Aproximación de la matriz 4225x4225 por métodos directos	37
17.	Error obtenido en el desarrollo de la matriz de 4225x4225	37

Índice de cuadros

1.	Tabla de valores iniciales para la función (1)	17
2.	Tabla de valores iniciales para la función (2)	17
3.	Tabla de resultados para la función (1)	20
4.	Tabla de resultados para punto fijo con y sin optimización (2)	20
5.	Ranking de métodos con 50 % vs 50 % para la función 1	23
6.	Ranking de métodos con 30 % vs 70 % para la función 1	24
7.	Ranking de métodos con 90 % vs 10 % para la función 1	24
8.	Tabla de resultados para la función (2)	24
9.	Tabla de resultados para punto fijo con y sin optimización (2)	25
10.	Ranking de métodos con 50 % vs 50 % para la función 2	27
11.	Ranking de métodos con 30 % vs 70 % para la función 2	28
12.	Ranking de métodos con 90 % vs 10 % para la función 2	28
13.	Tabla de aproximaciones para sistema de ecuación no lineal - Función 1	28
14.	Tabla de errores obtenidos para sistema de ecuaciones no lineal - Función 1 .	29
15.	Tabla de resumen para matriz de 289x289.	32
16.	Tabla de de costos espaciales para matriz de 289x289	32
17.	Tabla de resumen para matriz de 1089x1089.	35
18.	Tabla de de costos computacionales para matriz de 1089x1089	35
19.	Tabla de resumen para matriz 4225x4225.	38
20.	Tabla de de costos espaciales para matriz de 4225x4225	38
21.	Tabla de mayor iteración	39
22.	Tabla de mayor iteración	40
23.	Tabla de mayor iteración	40
24.	Tabla de mayor iteración	41
25.	Tabla de mayor iteración	42
26.	Tabla de mayor iteración	43
27.	Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 1	43

28.	Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 2	44
29.	Ranking de métodos con 50 % vs 50 % para la matriz 289x289	45
30.	Ranking de métodos con 30 % vs 70 % para la matriz 289x289	46
31.	Ranking de métodos con 90 % vs 10 % para la matriz 289x289	46
32.	Tabla de de costos computacionales para matriz de 289x289	46
33.	Ranking de métodos con 50 % vs 50 % para la matriz 1089x1089	48
34.	Ranking de métodos con 30 % vs 70 % para la matriz 1089x1089	48
35.	Ranking de métodos con 90 % vs 10 % para la matriz 1089x1089	49
36.	Tabla de de costos computacionales para matriz de 289x289	49
37.	Ranking de métodos con 50 % vs 50 % para la matriz 4225x4225	50
38.	Ranking de métodos con 30 % vs 70 % para la matriz 4225x4225	50
39.	Ranking de métodos con 90 % vs 10 % para la matriz 4225x4225	51
40.	Tabla de de costos computacionales para matriz de 4225x4225	51

1. Introducción

En la actualidad todo gira en torno a los números y diversos cálculos matemáticos, pero para esto se usan distintos tipos de operaciones. Por lo mencionado anteriormente, 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.

Un buen uso de los métodos numéricos otorga una infinidad de habilidades, las cuales pueden ser ocupadas en el día a día. Sin embargo, el problema que ocurre es que la computadora no puede aprender ciertas habilidades, por el simple motivo de no tener la capacidad pensante del humano. 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, además de diseñar un algoritmo capaz de entregar un ranking de los métodos según eficiencia y eficacia.

1.1.1. Objetivos específicos

1. Implementar los métodos de sistema de ecuaciones no lineales : Bisección, Secante, Regular Falso, Punto Fijo, Newton-Raphson, Newton-Raphson Multivariable.
2. Realizar un análisis a las 2 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.

4. Realizar un análisis a 3 matrices de distintas dimensiones con los métodos de sistemas de ecuaciones lineales.
5. Realizar un ranking de eficiencia vs eficacia según un porcentaje de distribución para los metodos, un ranking para los metodos de ec. no lineales y otro para los metodos de ec. lineales

1.2. Herramientas

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

1. Para la realización del código fuente se ocupo Matlab R2018b.
2. Para la realización de la documentación del laboratorio se ocupo Latex - Texstudio.

1.3. Estructura del informe

Este informa consta de un capitulo donde se procede a describir cada uno de los métodos que se van a utilizar dentro del desarrollo del laboratorio numero 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, indicando cual es su eficiencia y eficacia en cada uno de ellos con sus respectivas tablas de datos y gráficos de comparación obtenidos. Para despues realizar un raking de los metodos segun su eficiencia vs eficacia. Por ultimo, se realiza un análisis de cada uno de ellos para luego pasar al ultimo capitulo 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. Se quizo crear un capítulo para describir los métodos, ya que se considera importante explicar como funcionales estos métodos en las funciones que se aplicarán mas adelante.

2.1. Error

En su mayoría las distintos procedimientos que se realizan con numero reales es imposible representarlos en su forma exacta. Los errores surgen del uso de aproximaciones para representar operaciones y cantidades matematicas exactas. Estas incluyen los errores de truncamiento y de redondeo.

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 .(Plaza,2007,p. 7)

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

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.

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

h

$$TiempoCicloRelojP1 = \frac{1}{2} \quad (3)$$

$$TiempoCicloRelojP2 = \frac{1}{15} \quad (4)$$

$$TiempoCicloRelojP3 = \frac{1}{2,5} \quad (5)$$

Además, se define la eficiencia o el costo computacional $C_{computacional}$, 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:

$$C_{computacional} = \sum_{i=1}^n x_+ + x_- + x_* + x_ / \quad (6)$$

Donde cada x representa:

- 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.(Plaza,2007,p. 24)

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 :

$$m = \frac{a + b}{2} \quad (7)$$

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 :

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)} \quad (8)$$

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 :

$$x_r = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, k \geq 1 \quad (9)$$

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 líneas secantes y así aproximarse de manera mas certera a la raíz de la función $f(x)$.

La ecuación del método es:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, n \geq 1 \quad (10)$$

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:(Plaza,2007,p. 27)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n \geq 1 \quad (11)$$

2.2.2.3. Método de Newton para varias variables

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

$$\left\{ \begin{array}{lll} f_1(x_1, x_2, \dots, x_n) & = f_1(x) & = 0 \\ f_2(x_1, x_2, \dots, x_n) & = f_2(x) & = 0 \\ \vdots & \vdots & \vdots \\ f_n(x_1, x_2, \dots, x_n) & = f_n(x) & = 0 \end{array} \right.$$

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:

$$x^{k+1} = x^k + H^{(k)} \quad (12)$$

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:

$$Ax = b \rightarrow x = bA^{-1} \quad (13)$$

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 la matriz A , como el producto de una matriz triangular inferior L , con una matriz triangular superior U . Mediante el sistema de ecuaciones ya definido como :

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

Donde A se puede descomponer de la siguiente forma :

$$A = LU \quad (15)$$

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:

$$LUx = b \quad (16)$$

Esta descomposición nos sirve para poder resolver sistemas de ecuaciones lineales, a través de distintos métodos, entre ellos tenemos: Crout (no mencionado, ni utilizado en este laboratorio), Doolitte y Choslesky, los cuales veremos a continuación.

2.3.1.2. 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 raices 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:

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

2.3.1.3. 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:

$$Ax = b \quad (20)$$

La función anterior la descomponemos de la siguiente manera:

$$A = LL^T \quad (21)$$

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

$$Ly = b \quad (22)$$

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

Finalmente para obtener las soluciones de x, tenemos que:

$$L^T x = y \quad (23)$$

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

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

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

2.3.1.4. 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:

$$u_1 = v_1 \tag{24}$$

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

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

2.3.2. Métodos indirectos

También llamados metodos de descomposición, consisten en descomponer la matriz invertible A del sistema $Ax = b$ de la forma $A = M - N$. 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 :

$$Ax = B \quad (26)$$

Se descompone la matriz de A, de tal forma :

$$A = D + R \quad (27)$$

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:

$$Dx + Rx = b \quad (28)$$

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

$$x = D^{-1}b - D^{-1}Rx \quad (29)$$

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

$$x_i^{k+1} = D^{-1}b - D^{-1}Rx^k \quad (30)$$

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

$$x_i^{k+1} = \frac{1}{a_{ii}}(b_i - \sum_{i \neq j} r_{ij}x_j^k, i = 1, 2, 3, 4, 5, \dots) \quad (31)$$

2.3.2.2. Método de Gauss-Seidel

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

$$Ax = b \quad (32)$$

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

$$R = A - Q \quad (33)$$

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

$$Qx^k = b - Rx^{k+1} \quad (34)$$

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

$$Qx^k = \frac{(b_i - \sum_{j \neq i}^{i-1} a_{ij}x_j^k - \sum_{j \neq i+1}^n a_{ij}x_j^{k-1})}{a_{ii}}, i = 1, 2, 3, 4, 5, \dots \quad (35)$$

3. Funciones

3.1. Funciones para ecuaciones no lineales

Las funciones para la soluciones de ecuaciones no lineales del laboratorio son las siguientes :

- Función 1:

$$2x - 2^{-x} \quad (36)$$

- Función 2:

$$x^3 - 7x^2 + 14x - 2 \quad (37)$$

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

- Bisección
- Secante
- Regular Falsi
- Punto Fijo
- Newton-Raphson

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

$$E = 10^{-15} \quad (38)$$

Los puntos iniciales escogidos para cada una de las funciones se pueden observar en la Cuadro 1, Cuadro 2. Para los metodos del punto fijo y de Newton-Raphson se utiliza la Regla de Fourier para seleccionar el punto.

$f(x) = 2x - 2^{-x}$		
Método	x_0	x_1
Bisección	0	2
Secante	0	2
Regular Falsi	0	2
Punto Fijo	0	-
Newton-Raphson	0	-

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

$f(x) = x^3 - 7x^2 + 14x - 2$		
Método	x_0	x_1
Bisección	0	1
Secante	0	1
Regular Falsi	0	1
Punto Fijo	0	-
Newton-Raphson	0	-

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

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

- Función 1 :

$$\left\{ \begin{array}{l} x_1^2 - 10x_1 + x_2^2 + 8 = 0 \\ x_1x_2^2 + x_1 - 10x_2 + 8 = 0 \\ X_{(0)} = (0, 0)^T \end{array} \right.$$

- Función 2 :

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

El cual se ocupó un error para evaluar cada una de las funciones de:

$$E = 10^{-15} \tag{39}$$

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:

- Gauss.Jacobi.
- Gauss.Seidel.

- LU.
- Cholesky.
- QR.

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 2 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 $x_0 = 0$.

Se presenta la aproximación o valor, la eficiencia (costo computacional), la eficacia (error), el número de iteraciones .

Método	Aproximación	Error	Iteraciones	Costo Computacional
Bisección	0,38333234	$5,7 \cdot 10^{-16}$	53	256
Secante	0,38333234	0	7	61
Regula Falsi	0,38333234	0	7	89
Newton-Raphson	0,38333234	0	5	35

Cuadro 3: Tabla de resultados para la función (1)

Al método del punto fijo se le aplican la aceleración de Aitken y Steffensen. En la siguiente tabla se puede apreciar la menor cantidad de iteraciones que realiza el método del punto fijo aplicándole estas dos optimizaciones.

Método	Aproximación	Error	Iteraciones	Costo Computacional
Punto Fijo	0,38333234	$2,8 \cdot 10^{-16}$	28	56
Punto Fijo con Aitken	0,38333234	$1,4 \cdot 10^{-16}$	5	30
Punto Fijo con Steffensen	0,38333234	0	10	32

Cuadro 4: Tabla de resultados para punto fijo con y sin optimización (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.

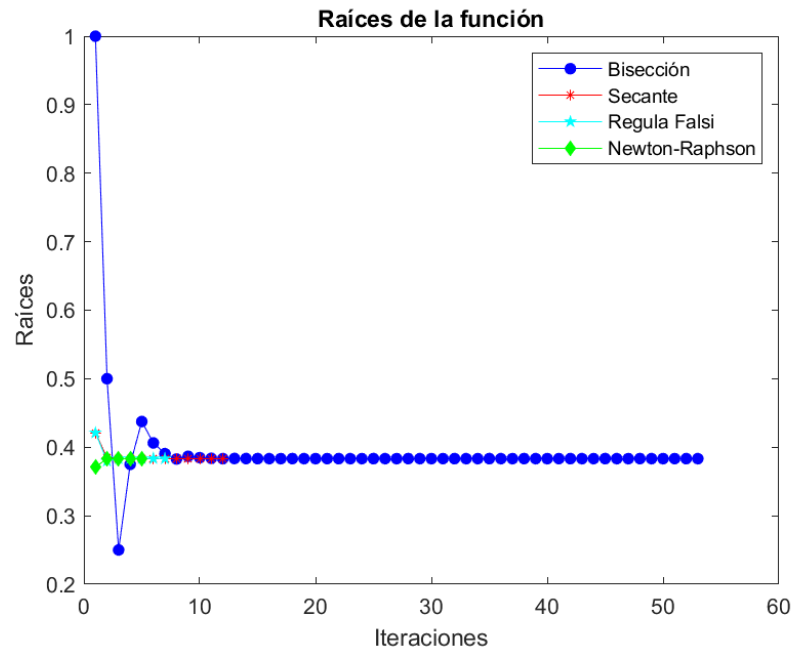


Figura 1: Resultado de las aproximaciones de la función 1

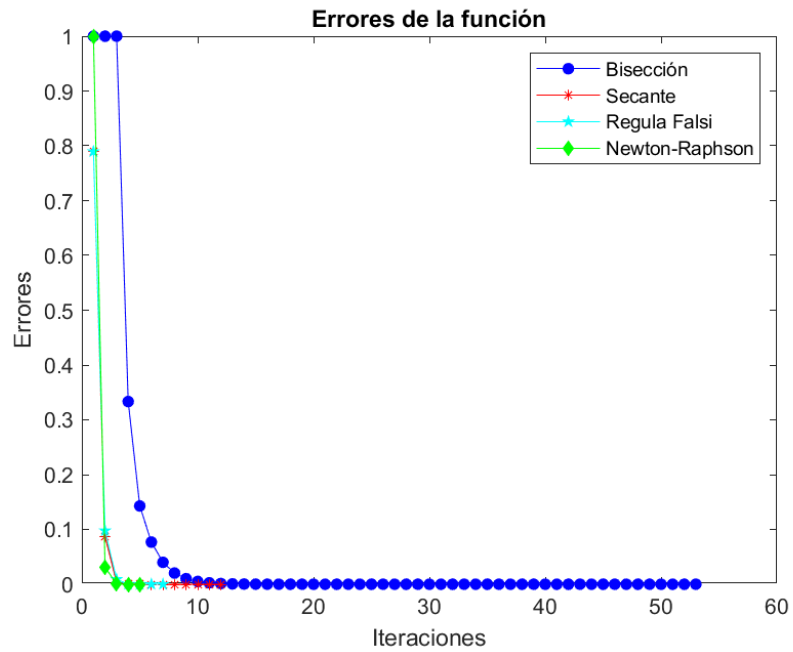


Figura 2: Resultados de los errores obtenidos en la función 1

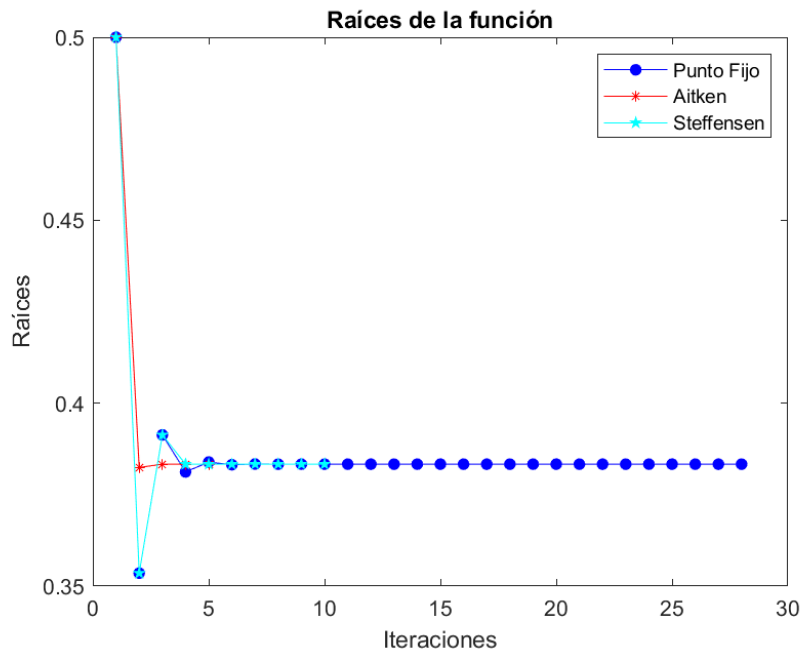


Figura 3: Resultado de las aproximaciones de la funcion 1 con método del punto fijo

■ Ranking de métodos

Se realiza un ranking de los métodos según eficiencia (costo computacional) y eficacia

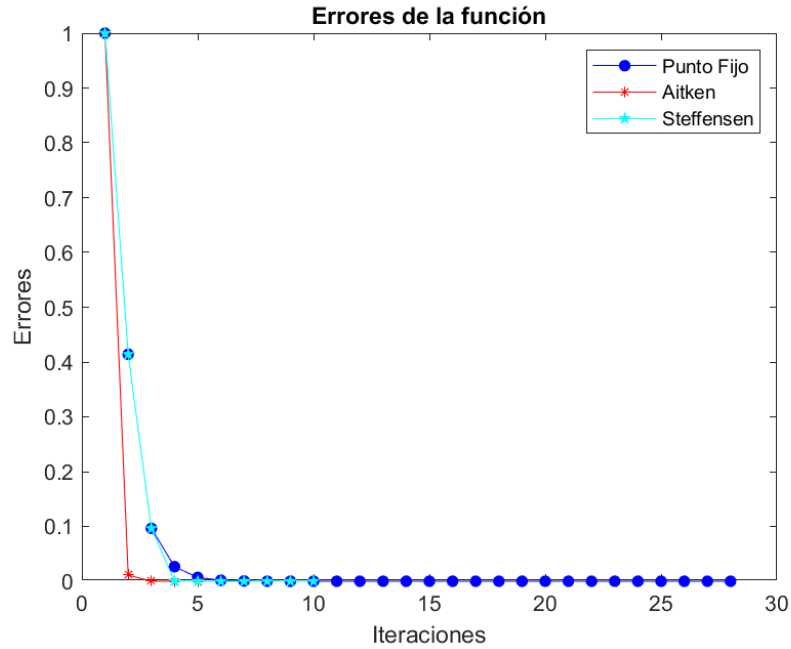


Figura 4: Resultados de los errores obtenidos en la función 1 con método del punto fijo

(precisión). Para ello se da un porcentaje de distribución, ejemplo: 50\$ vs 50 % no existe prioridad por ninguna medida, pero si es 30 % vs 70 % existe prioridad por la eficacia que por el costo. A continuación se darán tres ranking segun distinta prioridad.

Puesto	Método
1	Newton-Raphson
2	Secante
3	Punto Fijo
4	Regula Falsi
5	Bisección

Cuadro 5: Ranking de métodos con 50 % vs 50 % para la función 1

- Función $f(x) = x^3 - 7x^2 + 14x - 2$:

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 $x_0 = 0$.

Puesto	Método
1	Newton-Raphson
2	Secante
3	Regula Falsi
4	Punto Fijo
5	Bisección

Cuadro 6: Ranking de métodos con 30 % vs 70 % para la función 1

Puesto	Método
1	Newton-Raphson
2	Secante
3	Regula Falsi
4	Punto Fijo
5	Bisección

Cuadro 7: Ranking de métodos con 90 % vs 10 % para la función 1

Se presenta la aproximación o valor, la eficiencia (costo computacional), la eficacia (error), el número de iteraciones .

Método	Aproximación	Error	Iteraciones	Costo Computacional
Bisección	0,1545339	$7,1 \cdot 10^{-16}$	53	272
Secante	0,1545339	0	8	69
Regula Falsi	0,1545339	$1,7 \cdot 10^{-16}$	16	117
Newton-Raphson	0,1545339	0	6	42

Cuadro 8: Tabla de resultados para la función (2)

Al método del punto fijo se le aplican la aceleración de Aitken y Steffensen. En la siguiente tabla se puede apreciar la menor cantidad de iteraciones que realiza el método del punto fijo aplicándole estas dos optimizaciones.

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

Método	Aproximacion	Error	Iteraciones	Costo Computacional
Punto Fijo	0,1545339	$5,3 \cdot 10^{-16}$	19	38
Punto Fijo con Aitken	0,1545339	0	5	30
Punto Fijo con Steffensen	0,1545339	0	7	22

Cuadro 9: Tabla de resultados para punto fijo con y sin optimización (2)

los 4 métodos ocupados para esta sección del laboratorio.

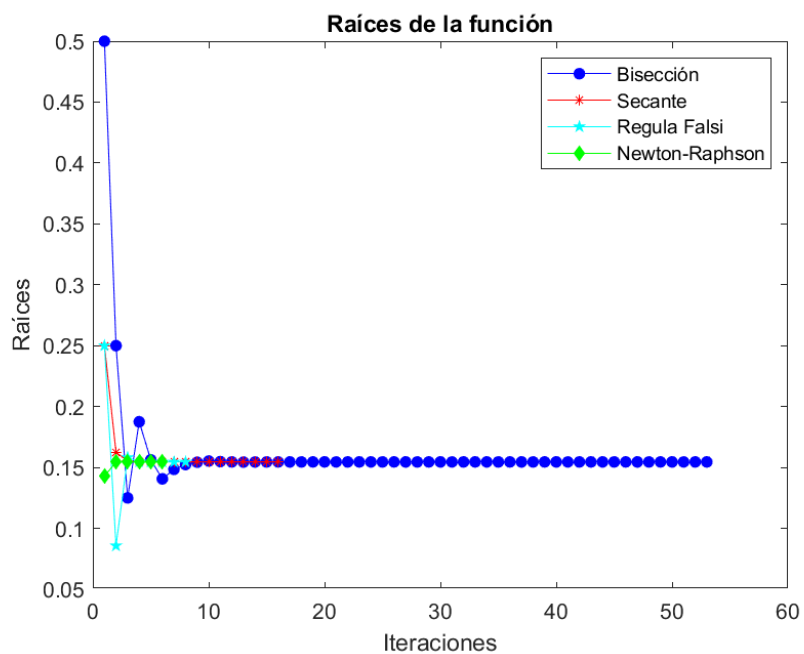


Figura 5: Resultado de las aproximaciones de la función 2

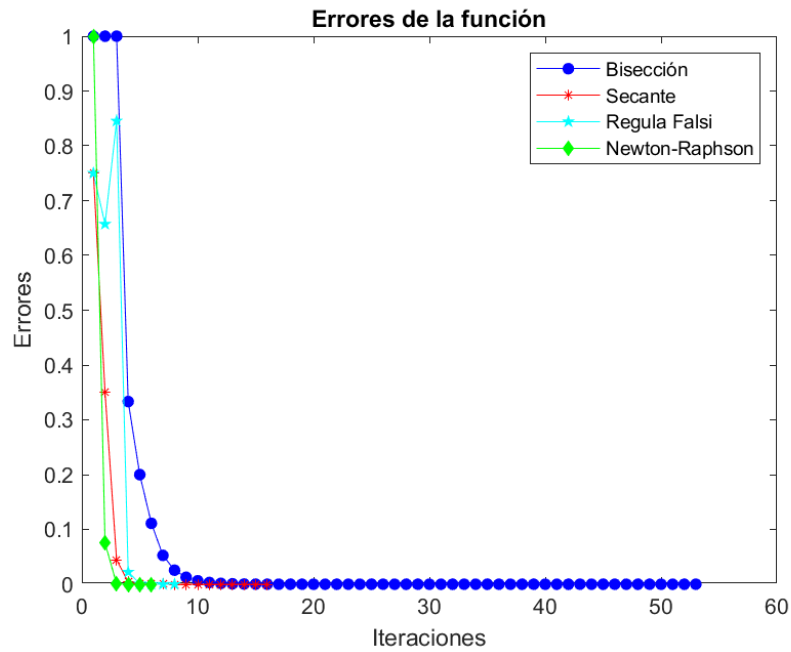


Figura 6: Resultados de los errores obtenidos en la función 2

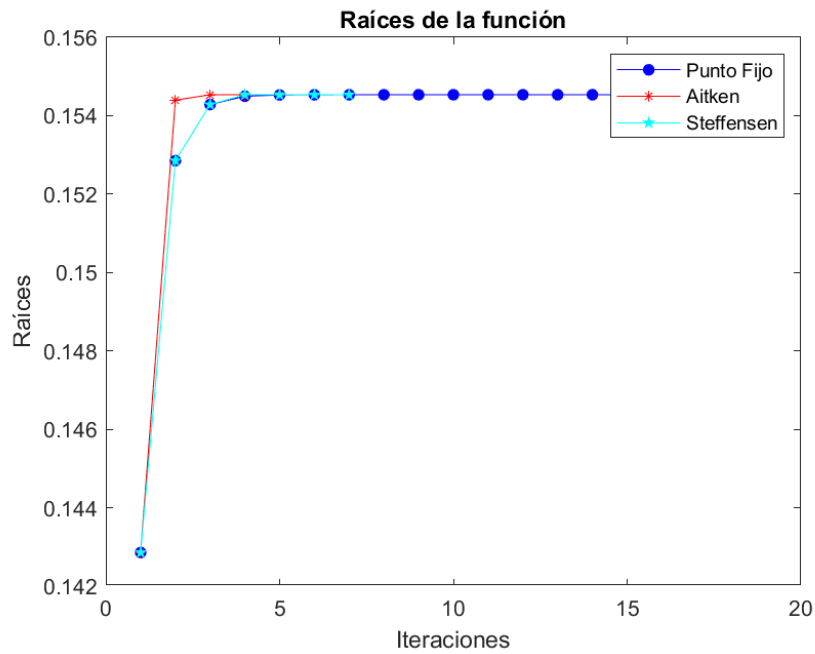


Figura 7: Resultado de las aproximaciones de la funcion 2 con método del punto fijo

■ Ranking de métodos

Se realiza un ranking de los métodos según eficiencia (costo computacional) y eficacia

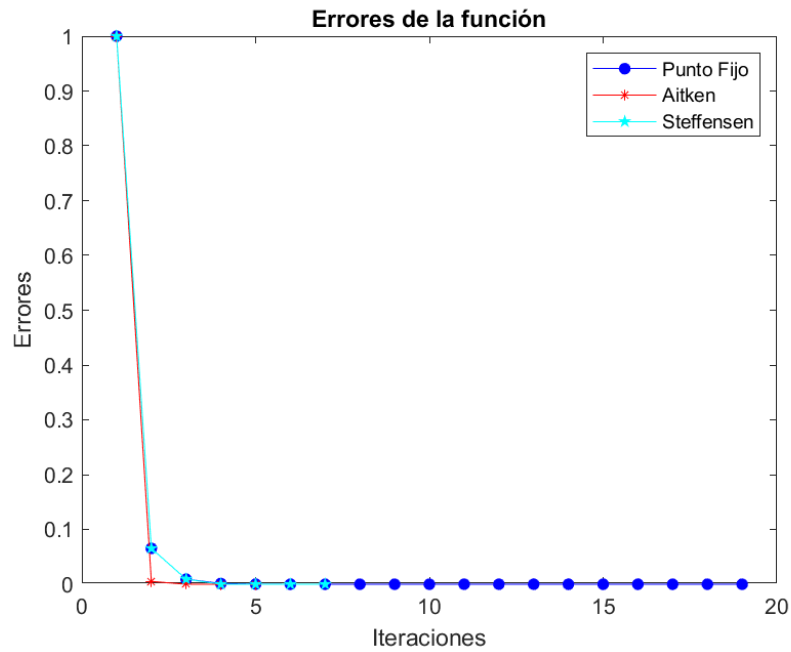


Figura 8: Resultados de los errores obtenidos en la función 2 con método del punto fijo

(precisión). Para ello se da un porcentaje de distribución, ejemplo: 50\$ vs 50 % no existe prioridad por ninguna medida, pero si es 30 % vs 70 % existe prioridad por la eficacia que por el costo. A continuación se darán tres ranking según distinta prioridad.

Puesto	Método
1	Newton-Raphson
2	Punto Fijo
3	Secante
4	Regula Falsi
5	Bisección

Cuadro 10: Ranking de métodos con 50 % vs 50 % para la función 2

Puesto	Método
1	Newton-Raphson
2	Punto Fijo
3	Secante
4	Regula Falsi
5	Bisección

Cuadro 11: Ranking de métodos con 30 % vs 70 % para la función 2

Puesto	Método
1	Punto Fijo
2	Newton-Raphson
3	Secante
4	Regula Falsi
5	Bisección

Cuadro 12: Ranking de métodos con 90 % vs 10 % para la función 2

4.2. Resultado de sistema de ecuaciones no lineal

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

- Función 1 :

$$\left\{ \begin{array}{l} x_1^2 - 10x_1 + x_2^2 + 8 = 0 \\ x_1x_2^2 + x_1 - 10x_2 + 8 = 0 \\ X_{(0)} = (0,0)^T \end{array} \right.$$

Variable	Aproximación
x_1	1
x_2	1

Cuadro 13: Tabla de aproximaciones para sistema de ecuación no lineal - Función 1

$$CostoComputacional = 60 \quad (40)$$

Se presenta el costo computacional como el de ambas variables, ya que se considera el calculo de todo el algoritmo, se incluye el calculo de las sumas, restas, multiplicaciones y divisiones.

Iteraciones	Error
1	1
2	0.153010368
3	0.008222630
4	2.812301703e-5
5	3.458923147e-10
6	0

Cuadro 14: Tabla de errores obtenidos para sistema de ecuaciones no lineal - Función 1

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 :

$$E = 10^{-15} \quad (41)$$

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:

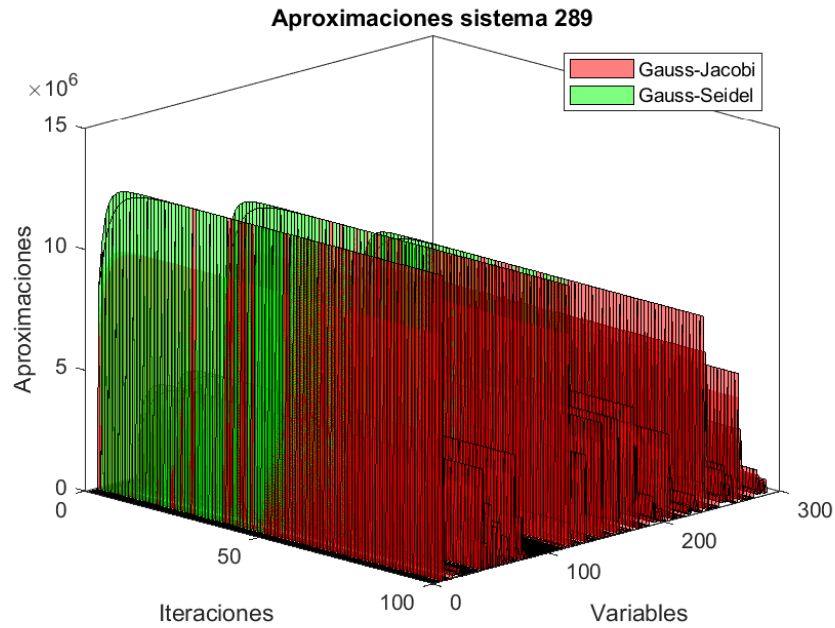


Figura 9: Aproximación de la matriz 289x289 por métodos iterativos

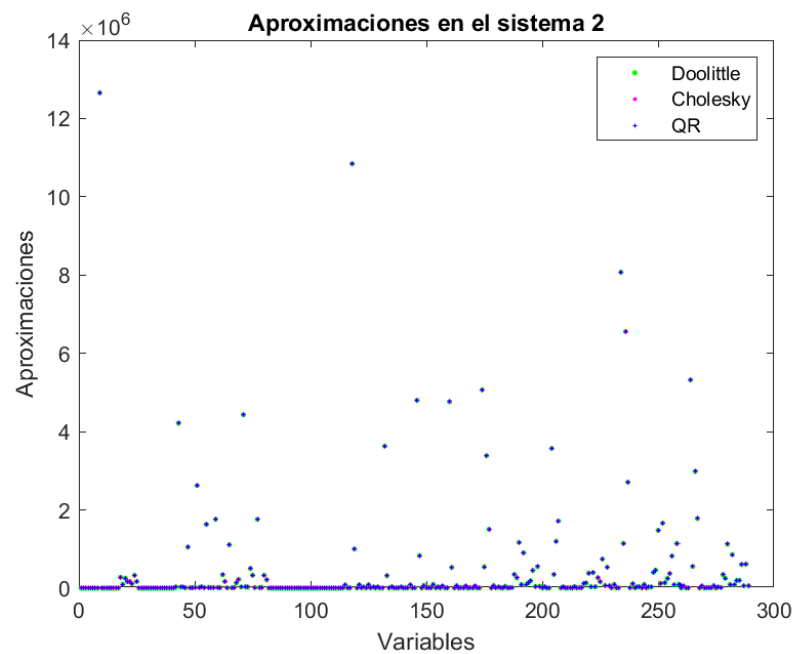


Figura 10: Aproximación de la matriz 289x289 por métodos directos

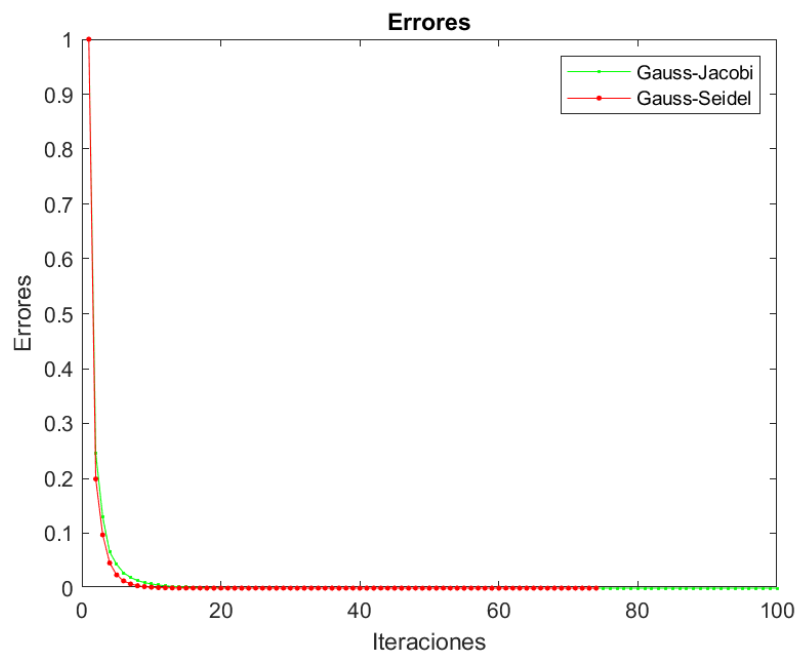


Figura 11: Error obtenido en el desarrollo de la matriz de 289x289

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:

Método	Iteraciones	Error
Gauss-Jacobi	100	$6,2308795e^{-12}$
Gauss-Seidel	74	$5,87434439e^{-16}$
Doolittle	-	$2,206155357251e^{-15}$
Cholesky	-	$3,15106810710e^{-15}$
QR	-	$2,9114751231e^{-15}$

Cuadro 15: Tabla de resumen para matriz de 289x289.

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

Método	Costos computacionales
Gauss-Jacobi	889
Gauss-Seidel	881
Doolittle	16.049.905
Cholesky	83.811
QR	125.140

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

■ Matriz 1089

Las dimensiones de esta matriz recaen en 1089×1089 , 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:

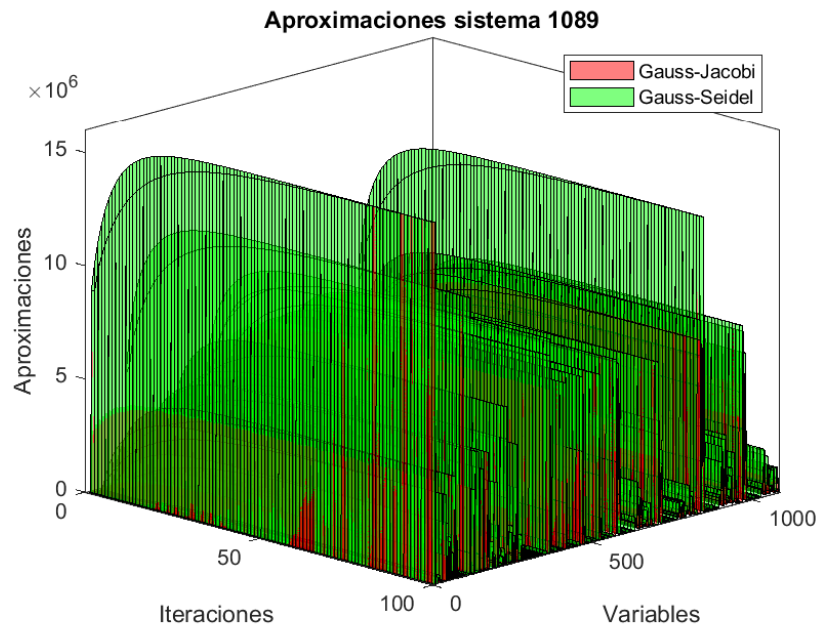


Figura 12: Aproximación de la matriz 1089×1089 por métodos iterativos

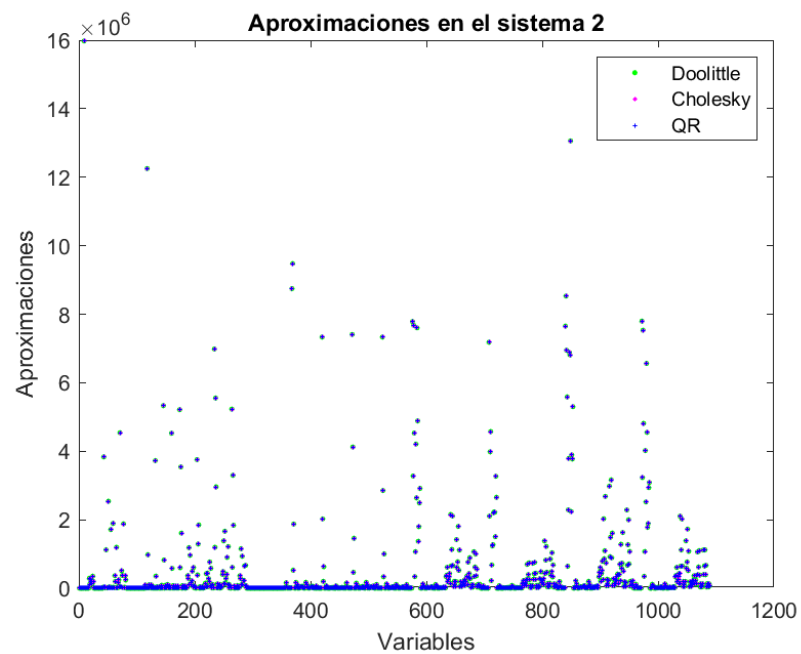


Figura 13: Aproximación de la matriz 1089x1089 por métodos directos

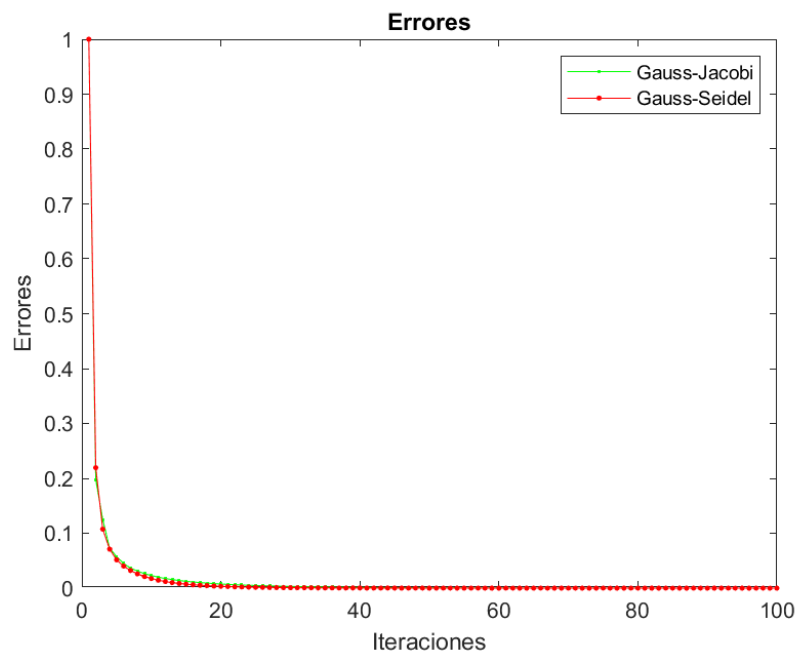


Figura 14: Error obtenido en el desarrollo de la matriz de 1089x1089

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:

Método	Iteraciones	Error
Gauss-Jacobi	100	$3,077712757e^{-05}$
Gauss-Seidel	100	$2,210626159e^{-07}$
Doolittle	-	$13,5497480578858e^{-14}$
Cholesky	-	$5,283192595769840e^{-14}$
QR	-	$1,180438366378053e^{-14}$

Cuadro 17: Tabla de resumen para matriz de 1089x1089.

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

Método	Costos computacionales
Gauss-Jacobi	1.689
Gauss-Seidel	1.889
Doolittle	860.385.505
Cholesky	1.187.012
QR	1.778.342

Cuadro 18: Tabla de de costos computacionales para matriz de 1089x1089

- Matriz 4225

Las dimensiones de esta matriz recaen en 4225 x 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:

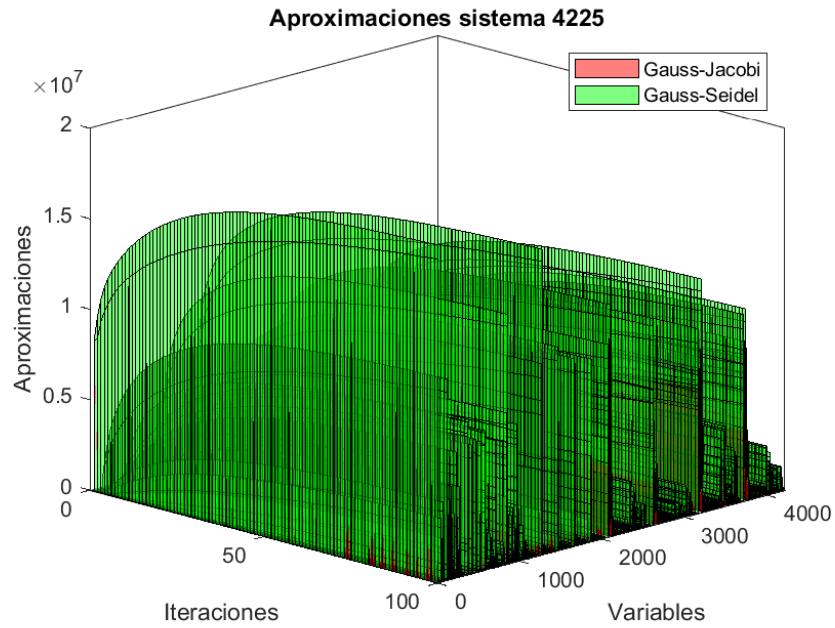


Figura 15: Aproximación de la matriz 4225x4225 por métodos iterativos

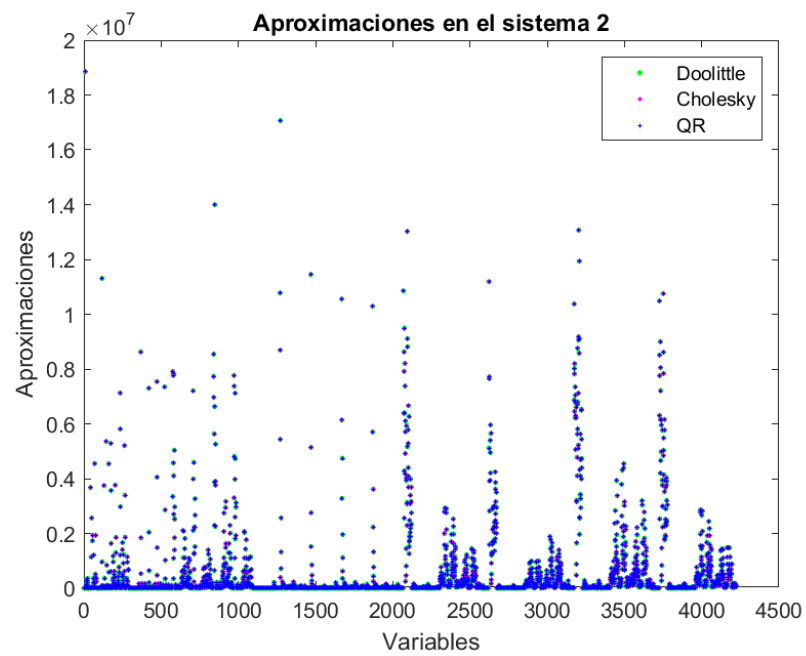


Figura 16: Aproximación de la matriz 4225x4225 por métodos directos

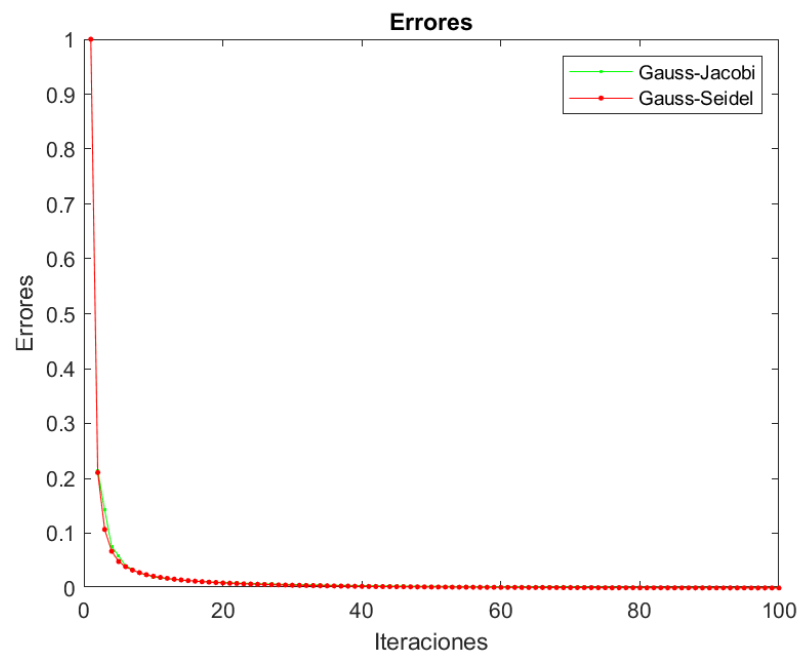


Figura 17: Error obtenido en el desarrollo de la matriz de 4225x4225

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
Gauss-Jacobi	100	0,001008438291151
Gauss-Seidel	100	$3,17222444411e^{-04}$
Doolittle	-	$65,130340483177260e^{-14}$
Cholesky	-	$27,307890152161573e^{-14}$
QR	-	$,407352972760017e^{-14}$

Cuadro 19: Tabla de resumen para matriz 4225x4225.

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

Método	Costos computacionales
Gauss-Jacobi	4.825
Gauss-Seidel	5.025
Doolittle	$5,027e^{10}$
Cholesky	17.854.851
QR	26.773.828

Cuadro 20: Tabla de de costos espaciales para matriz de 4225x4225

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 analizara 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 se deja en claro es que esta función consta solamente de una raíz, la cual se produce cuando $x = 2$. Por ende, no tenemos complicaciones de saber cual es la raíz que se esta 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 numero de iteraciones quedando de la siguiente forma :

Método	Iteraciones
Bisección	53
Regular Falsi	7
Secante	7
Newton-Raphson	5

Cuadro 21: Tabla de mayor iteración

Esto nos indica que el método que necesita más iteraciones para llegar a la raíz es Bisección y el método que se acerca en menos iteraciones a la raíz es el método de Newton-Raphson.

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 destacar un detalle que el método Regular Falsi y Secante y Newton-Raphson tienen una diferencia muy menor al momento de alcanzar el eje de las abscisas, comparado con el método de la Bisección.

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

Método	Costo computacional
Newton-Raphson	5
Secante	8
Regular Falsi	13
Bisección	53

Cuadro 22: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor costo computacional es Bisección y por el otro extremo el que produce un menor costo computacional es el método de Newton-Raphson. Esto se puede realizar con que con mas iteraciones mayor seria el costo computacional, aunque no siempre es asi, esto se explicará en los ranking de los métodos.

Ahora se analiza el Metodo del punto fijo. El primer cuando de la tabla 26. se realiza la ejecución del algoritmo sin metodos de optimización, despues muestra la cantidad de iteraciones con Aitken y la otra con Steffensen:

Método	Iteraciones
Punto Fijo	28
Punto Fijo con Aitken	5
Punto Fijo con Steffensen	10

Cuadro 23: Tabla de mayor iteración

Como se puede apreciar que el método del Punto Fijo produce un mayor numero de iteraciones cuando no se implementa ningun método de aceleración de convergencia. Cuando

se implementan estos métodos las iteraciones disminuyen, es decir se llega a la raíz mas rápido (aceleración de convergencia), todo esto se puede gracias a la regla de Fourier que escoge el punto inicial para garantizar convergencia.

Al analizar las tablas de ranking según un porcentaje de distribución podemos analizar que cuando se le da prioridad a la eficiencia o a la eficacia el método de Regula Falsi, sube al tercer puesto, mientras que el método del punto fijo (sin ningún método de optimización) baja al cuarto puesto, esto se debe a que el método de punto fijo tiene menor costo computacional que el otro método.

5.1.2. Función 2 : $f(x) = x^3 - 7x^2 + 14x - 2$

Lo primero que tenemos que dejar en claro que esta función consta de una raíz, aunque tomaremos un intervalo definida por $[0, 1]$, el cual consta con esta raíz, por ende, sabremos que raíz se está 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 número de iteraciones quedando de la siguiente forma :

Método	Iteraciones
Bisección	53
Regular Falsi	16
Secante	8
Newton-Raphson	6

Cuadro 24: 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 es Newton-Raphson.

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

- Bisección

- Secante
- Regular Falsi
- Newton-Raphson

Como se puede apreciar que el método que produce un mayor costo computacional es Bisección y por el otro extremo el que produce un menor costo computacional es el método de Newton-Raphson. Esto se puede realizar con que con mas iteraciones mayor seria el costo computacional, aunque no siempre es asi, esto se explicará en los ranking de los métodos.

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

Método	Costo computacional
Newton-Raphson	6
Secante	9
Regular Falsi	17
Bisección	53

Cuadro 25: Tabla de mayor iteración

Como se puede apreciar que el método que produce un mayor costo computacional es Bisección y por el otro extremo el que produce un menor costo computacional es el método de Newton-Raphson. Esto se puede realizar con que con mas iteraciones mayor seria el costo computacional, aunque no siempre es asi, esto se explicará en los ranking de los métodos.

Ahora se analiza el Metodo del punto fijo. El primer cuando de la tabla 29. se realiza la ejecución del algoritmo sin metodos de optimización, despues muestra la cantidad de iteraciones con Aitken y la otra con Steffensen:

Como se puede apreciar que el método del Punto Fijo produce un mayor numero de iteraciones cuando no se implementa ningun método de aceleración de convergencia. Cuando se implementan estos métodos las iteraciones disminuyen, es decir se llega a la raíz mas rápido (aceleracion de convergencia), todo esto se puede gracias a la regla de Fourier que escoge el punto inicial para garantizar convergencia.

Método	Iteraciones
Punto Fijo	19
Punto Fijo con Aitken	5
Punto Fijo con Steffensen	7

Cuadro 26: Tabla de mayor iteración

Al analizar las tablas de ranking según un porcentaje de distribución podemos analizar que cuando se le da prioridad a la eficiencia el método del Punto Fijo sube al primer puesto, esto se debe a que a pesar de realizar mayores iteraciones que el método de Newton, realiza menos operaciones en el algoritmo. Cuando se prioriza eficacia el ranking se mantiene igual ya que todos tienen precisiones bastante distintas (muy distantes entre cada método) a los otros métodos, entonces al priorizar ese porcentaje no afecta el ranking.

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 expresa como $E = 0$.

El análisis de la siguiente tabla consiste en :

Tiempo	Costos espacial (+ y -)	Costos espacial (/ y *)
0,4354570000000000	10	5

Cuadro 27: Tabla de costo espacial y tiempo de ejecución para sistema de ecuación no lineal - Función 1

Se puede afirmar que el método 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 28: 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 el 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 oportunidades 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 los gráficos entregados en el capítulo 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 18, tenemos que el método Gauss-Jacobi realizó un total de 26 iteraciones más que el método Gauss-Seidel.

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

Por último, se procede a entregar las tablas de los rankings de los métodos, según un porcentaje de distribución.

Puesto	Método
1	Gauss.Seidel
2	Gauss.Jacobi
3	Cholesky
4	QR
5	Doolittle

Cuadro 29: Ranking de métodos con 50 % vs 50 % para la matriz 289x289

Podemos analizar que variando el porcentaje de distribución no afecta el ranking, esto ocurre porque la diferencia entre los costos y la diferencia entre la precisión de los métodos es muy distante, entonces realizar preferencias hacia algún lado no afecta en nada el top final.

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

Puesto	Método
1	Gauss.Seidel
2	Gauss.Jacobi
3	Cholesky
4	QR
5	Doolittle

Cuadro 30: Ranking de métodos con 30 % vs 70 % para la matriz 289x289

Puesto	Método
1	Gauss.Seidel
2	Gauss.Jacobi
3	Cholesky
4	QR
5	Doolittle

Cuadro 31: Ranking de métodos con 90 % vs 10 % para la matriz 289x289

Método	Costos Computacionales
Gauss-Jacobi	889
Gauss-Seidel	881
Doolittle	16.049.905
Cholesky	83.811
QR	125.140

Cuadro 32: Tabla de de costos computacionales 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 computacionales que realizan. Además, se destaca que 'Gauss-Seidel' tiene un mejor rendimiento que 'Gauss-Jacobi', aunque tenga un costo computacional menor, ya que tiene menor costo y menos cantidad de operaciones.

Además, podemos afirmar que el método que mas costo computacional tiene es el método de Doolittle, superando por mucho a los demás métodos expuestos en el cuadro 33. Además, tenemos que el método Cholesky tiene un menor costo computacional, lo cual lo califica como el mejor método directo evaluándolo desde el punto de vista del costo computacional.

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 ultimo, se procede a entregar las tablas de los ranking de los metodos, segun un porcentaje de distribucion.

Puesto	Método
1	Gauss.Jacobi
2	Gauss.Seidel
3	Cholesky
4	QR
5	Doolittle

Cuadro 33: Ranking de métodos con 50 % vs 50 % para la matriz 1089x1089

Puesto	Método
1	Gauss.Jacobi
2	Gauss.Seidel
3	Cholesky
4	QR
5	Doolittle

Cuadro 34: Ranking de métodos con 30 % vs 70 % para la matriz 1089x1089

Podemos analizar que variando el porcentaje de distribución no afecta el ranking, esto ocurre porque la diferencia entre los costos y las diferencia entre la precisión de los métodos es muy distante, entonces realizar preferencias hacia algún lado no afecta en nada

Puesto	Método
1	Gauss.Jacobi
2	Gauss.Seidel
3	Cholesky
4	QR
5	Doolittle

Cuadro 35: Ranking de métodos con 90 % vs 10 % para la matriz 1089x1089

el top final. Aunque nos podemos fijar que el primer puesto varió con respecto a la matriz anterior, ahora el puesto 1 lo tiene Gauss.Jacobi, esto se debe a que aumento la cantidad de iteraciones del metodo Gauss.Seidel con relacion a la matriz anterior y su costo computacional ahora es mayor que en la matriz anterior.

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

Método	Costos Computacionales
Gauss-Jacobi	889
Gauss-Seidel	881
Doolittle	16.049.905
Cholesky	83.811
QR	125.140

Cuadro 36: Tabla de de costos computacionales 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 computacionales que realizan. Además, se destaca que 'Gauss-Jacobi' tiene un mejor rendimiento que 'Gauss-Seidel'.

Además, podemos afirmar que el método que mas costo computacional tiene es el método de Doolitte, superando por mucho a los demás métodos expuestos en el cuadro 33. Además, tenemos que el método Cholesky tiene un menor costo computacional, lo cual lo califica como el mejor método directo evaluándolo desde el punto de vista del costo computacional.

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 ultimo, se procede a entregar las tablas de los ranking de los metodos, segun un porcentaje de distribucion.

Puesto	Método
1	Gauss.Jacobi
2	Gauss.Seidel
3	Cholesky
4	QR
5	Doolittle

Cuadro 37: Ranking de métodos con 50 % vs 50 % para la matriz 4225x4225

Puesto	Método
1	Gauss.Jacobi
2	Gauss.Seidel
3	Cholesky
4	QR
5	Doolittle

Cuadro 38: Ranking de métodos con 30 % vs 70 % para la matriz 4225x4225

Podemos analizar que variando el porcentaje de distribución no afecta el ranking, esto ocurre porque la diferencia entre los costos y las diferencia entre la precisión de los métodos es muy distante, entonces realizar preferencias hacia algún lado no afecta en nada

Puesto	Método
1	Gauss.Jacobi
2	Gauss.Seidel
3	Cholesky
4	QR
5	Doolittle

Cuadro 39: Ranking de métodos con 90 % vs 10 % para la matriz 4225x4225

el top final. Aunque nos podemos fijar que el primer puesto varió con respecto a la matriz anterior, ahora el puesto 1 lo tiene Gauss.Jacobi, esto se debe a que aumento la cantidad de iteraciones del metodo Gauss.Seidel con relacion a la matriz anterior y su costo computacional ahora es mayor que en la matriz anterior.

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

Método	Costos Computacionales
Gauss-Jacobi	4.825
Gauss-Seidel	5.025
Doolittle	$5,025e^{10}$
Cholesky	17.854.851
QR	26.773.828

Cuadro 40: Tabla de de costos computacionales 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 computacionales que realizan. Además, se destaca que 'Gauss-Jacobi' tiene un mejor rendimiento que 'Gauss-Seidel'.

Además, podemos afirmar que el método que mas costo computacional tiene es el método de Doolitte, superando por mucho a los demás métodos expuestos en el cuadro 43. Además, tenemos que el método Cholesky tiene un menor costo computacional, lo cual lo califica como el mejor método directo evaluándolo desde el punto de vista del costo computacional.

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.

También mencionar la implementación de la aceleración de convergencia de Aitken y Steffensen en el método del punto fijo, nos sirve para analizar de como converge con mayor rapidez desde la iteración 2 en adelante.

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 computacional que se consideró consistía en la operaciones de realizaba el algoritmo, las operaciones básicas suma, resta, 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 realizar el método de Newton Multivariable sin usar funciones nativas de 'Matlab', se pudo calcular igual con exactitud las raíces de los sistemas de ecuaciones. 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 entregó herramientas y mayor entendimiento de cómo implementar una abstracción matemática a un computador.

7. Anexo

7.1. Métodos Numéricos

Como ya se describieron los métodos en el capítulo de Marco Teórico, se debe indicar que cada uno fue obtenido del libro de Sergio Plaza de Métodos Numéricos donde se explican con más exactitud los métodos.

7.2. Objetivos no cumplidos

Ya que no se pudo realizar el método de Newton Multivariable se incluye en la librería `Intento Multivariable`.^{en} la carpeta de NLS, esta es una carpeta del intento de los métodos. Además se realizó el método de Newton Multivariable para dos variables sin usar ninguna función nativa incluidas en 'Matlab'.

8. Bibliografía

- Plaza, S. (2007), Métodos Numéricos, Santiago De Chile.
- Rodríguez, L. (2011), Análisis numérico Básico, Guayaquil, Ecuador.