UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

INFORME DE LABORATORIO 1 DE ALGORITMOS NUMÉRICOS: MÉTODOS NUMÉRICOS

PATRICIA MELO

Profesor: Óscar Rojas Díaz.

Santiago - Chile

15 de mayo de 2019

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

CAPÍTULO 1. INTRODUCCIÓN

En el presente informe se muestra el desarrollo de 8 métodos numéricos en *MATLAB*, que son *Newton para varias variables, Gauss.Jacobi, Gauss.Seidel, LU, Cholesky, QR, Givens y Householder.*

Los métodos numéricos son técnicas o herramientas mediante las cuales es posible formular problemas de tal forma que sean resueltos con operaciones aritméticas. Aunque hay muchos tipos de métodos, todos comparten una característica común, llevan a cabo un buen número de cálculos aritméticos y emiten soluciones aproximadas. Estas herramientas permiten describir el comportamiento de distintos fenómenos físicos, químicos, e incluso sociales. La importancia de los métodos numéricos no radica en buscar la solución exacta de un problema, sino en entrgar una aproximada pero con la precisión requerida, es decir, con un error lo más próximo a cero, de ahí la utilidad de los métodos numéricos.

El objetivo principal de este laboratorio es analizar las características de los 8 métodos presentados, considerando la rapidez con que convergen a la solución, el tamaño del error de cada uno de ellos, los costos operacionales y los resultados obtenido, todo esto considerando las iteraciones realizadas. Para esto se programaron los algoritmos en MATLAB, consiguiendo un manejo básico de la programación en este lenguaje y utilizando los gráficos que puede realizar este programa para comparar los resultados obtenidos.

En este informe primero se abordará una pequeña descripción de los métodos, para así contextualizar el desarrollo del trabajo y entender cómo funciona, de manera general, cada uno de los algoritmos programados. Luego de esto se presenta el sistema de ecuaciones con que trabajará un método en particular, y la descripción de tres matrices, las cuales el resto de los métodos utilizarán. En el cuarto capítulo se presentará la parte de los resultados, donde se muestran los gráficos y tablas correspondientes. Para continuar, se presenta un análisis con los datos y gráficos obtenidos como resultados, para de esta forma poder comparar cada uno de los métodos. Finalmente, se presentan algunas conclusiones del trabajo realizado.

CAPÍTULO 2. DESCRIPCIÓN DE LOS MÉTODOS

En este capítulo se presenta una breve descripción de cada uno de los métodos, indicando además el procedimiento para aproximar las soluciones a las distintas ecuaciones.

2.1 MÉTODO DE NEWTON-RHAPSON

Este es un método eficiente para encontrar las raíces o aproximaciones a ceros de los sistemas de ecuaciones. Es un método abierto, que no garantiza su convergencia global. La única forma de alcanzar la convergencia es seleccionar un valor suficientemente cercano a la raíz buscada. Es importante mencionar que este método tiene dificultades cuando la función tiene pendientes grandes o múltiples puntos de inflexión cerca de la raíz, lo que trae como consecuencia un aumento de probabilidad de que no exista convergencia.

La fórmula de iteración para este método es la siguiente.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.1}$$

Donde se va calculando los números siguientes con la ecuación (2.1) hasta tener la precisión deseada.

2.2 MÉTODO ITERATIVO GAUSS.JACOBI

Es usado para resolver sistemas de ecuaciones lineales del tipo $Ax = b$, se basa en el punto fijo. Consiste principalmente en construir una sucesión convergente definida iterativamente. El algoritmo se detiene al llegar a la solución del sistema o hasta ucmplir con una cantidad finita de iteraciones, la cual puede entregar una aproximación del resultado. La sucesión se logra descompociendo la matriz A en

$$A = D + R \tag{2.2}$$

Donde D es una matriz diagonal y R la suma de la matriz inferior L con la matriz superior U, de esta forma finalmente se tiene x (solución) para cada iteración:

$$x = D^{-1}(b - Rx) \tag{2.3}$$

La matriz A converge siempre cuando ésta es dominante.

2.3 MÉTODO ITERATIVO GAUSS.SEIDEL

Al igual que Gauss.Jacobi, este método es utilizado para resolver sistemas de ecuaciones lineales del tipo $Ax = b$, de manera iterativa. Para que la matriz A converge, ésta debe ser dominante, positiva y simétrica. Para cada iteración i (de 1 hasta n) y $a_{ii} \neq 0$ se tiene:

$$x_i^{k+1} = \frac{-\sum_{1<=j<=i-1} a_{ij}x_j^{k+1} - \sum_{i+1<=j<=n} a_{ij}x_j^k + b_i}{a_{ii}} \tag{2.4}$$

Comparado con el método Gauss.Jacobbi, este es por lo menos igual de rápido, con respecto a la convergencia.

2.4 MÉTODO LU

Es un método directo de factorización, el cual descompone la matriz A en dos submatrices, donde $A = LU$. L es una matriz inferior y U una matriz superior, además se debe resolver el sistema $Ax = b$. Reemplazando A por las submatrices en el sistema tenemos:

$$Ax = (LU)x = L(Ux) \tag{2.5}$$

Una posible solución es resolver $y = Ux$, para luego tener $Ly = b$. Para lograr esa factorización en este laboratorio se utilizará Doolittle, donde la submatriz L queda con una diagonal de 1, y así se puede obtener la submatriz U.

2.5 MÉTODO CHOLESKY

También es un método directo, donde utiliza la factorización LU. Para utilizar este método se necesita que la matriz sea positiva y simétrica. Realiza apróximadamente la mitad de las operaciones que LU con Croust o Doolittle. Si se tiene que $Ax = b$, con Cholesky la matriz A quedaría como L^tL, de esta forma se trabaja resolviendo el siguiente sistema de ecuacines:

$$L^ty = b, Lx = y \tag{2.6}$$

2.6 MÉTODO QR

Es un método de factorización, la cual descompone una matriz A en una submatriz ortogonal de la misma y en otra submatriz triangular superior, de esta forma se tiene $A = QR$, donde Q es la matriz ortogonal y R la triangular superior. En esta entrega se utilizará el método de Gram-Schmidt, el cual se encarga de ortogonalizar con las columnas de A como vectores a procesar. Gram-Schmidt no es estable numéricamente hablado, ya que se ve afectado por los errores de redondeo en los vectores ortogonales.

2.7 MÉTODO GIVENS

El método Givens consiste en sucesivas transformaciones de rotación usando matrices de Givens hasta reducir la matriz A rectangular mxn a una triangular superior. Se suele usar este método para matrices dispersas. Las rotaciones de Gives se aplican a pares sucesivos de filas para conseguir hacer los ceros suficientes para obtener el factor R de la descomposicióon QR.

2.8 MÉTODO HOUSEHOLDER

Para este método se tiene una matriz H de la siguiente forma:

$$H = I - \frac{2}{v^t v} v v^t \tag{2.7}$$

Donde v es un vector no nulo, vv^t es una matriz cuadrada y $v^t v$ un número. La matriz H es ortogonal y simétrica, e I la matriz de identidad. Este método consiste en la sucesiva transformación de reflexiones utilizando matrices Householder, hasta reducir la matriz A a una matriz triangular. La diferencia que tiene Householder con respecto a Givens es que esta al premultiplicar A hacen que se anulen todos los elementos de la columna que necesite, no elemento a elemento.

CAPÍTULO 3. DESARROLLO

En este capítulo se describe con qué se trabajo para cada método.

3.1 PARA NEWTON MULTIVARIABLE

Para utilizar Newton-Raphson para varias variables se tiene el siguiente sistema de ecuaciones:

$$\begin{cases} \mathrm{x}_1^2 + x_2 - 37 = 0 \\ \mathrm{x}_1 - x_2^2 - 5 = 0 \\ \mathrm{x}_1 + x_2 + x_3 - 3 = 0 \\ \mathrm{X}_{(0)} = (0,0,0)^T \end{cases}$$

3.2 MATRICES

Para el resto de los métodos se trabaja con una matriz A cuadrada y un vector b. Los tamaños y características de las matrices y vectores son los siguientes:

1. Matriz 289x289: Tiene como característaicas ser definida positiva, semi definida positiva, simétrica, dominante y dispersa. Gracias a estas características todos los métodos pueden trabajar con ella.

2. Matriz 1089x1089: Esta matriz puede ser considerada mediana, sus características son semi definida positiva, dominante, dispersa y simétrica.

3. Matriz 4225x4225: Es una matriz mucho más grande que las anteriores, sus características son dispersa, dominate, semi definida positiva y simétrica. Existen 5 métodos que pueden trabajar con ella que se nombran más adelante.

4. Vecto b: El tamaño del vector varia segun la matriz, debe ser 1xn, con n como las filas de la matriz que le corresponde, de esta forma se tiene $Ax = b$.

Cabe destacar que las matrices y vectores fueron entregadas a través de moodle por el profesor. Para detectar las características de las matrices se crearon 6 funciones que verican si cumple o no con con dichas características.

CAPÍTULO 4. RESULTADOS

A continuación se presentan los gráficos creados a partir de los resultados obtenidos de la aplicación de los métodos propuestos, cabe mencionar que en algunos casos no se realizó la prueba completa de todos los métodos, principalmente por falta de tiempo o porque no cumplen las matrices con las características necesarias para ser trabajadas. A continunación se presenta primero los resultados obtenidos por Newton, seguido por el resto de los resultados de los métodos con las diferentes matrices descritas en el capítulo anterior, los gráficos y tablas serán una comparación entre tiempos, costos, resultados y errores.

4.1 UTILIZANDO NEWTON MULTIVARIABLE

Utilizando el sistema de ecuaciones anteriormente descrito se obtiene la siguiente gráfica con los resultados.

4.1.1 Resultados Newton multivariable

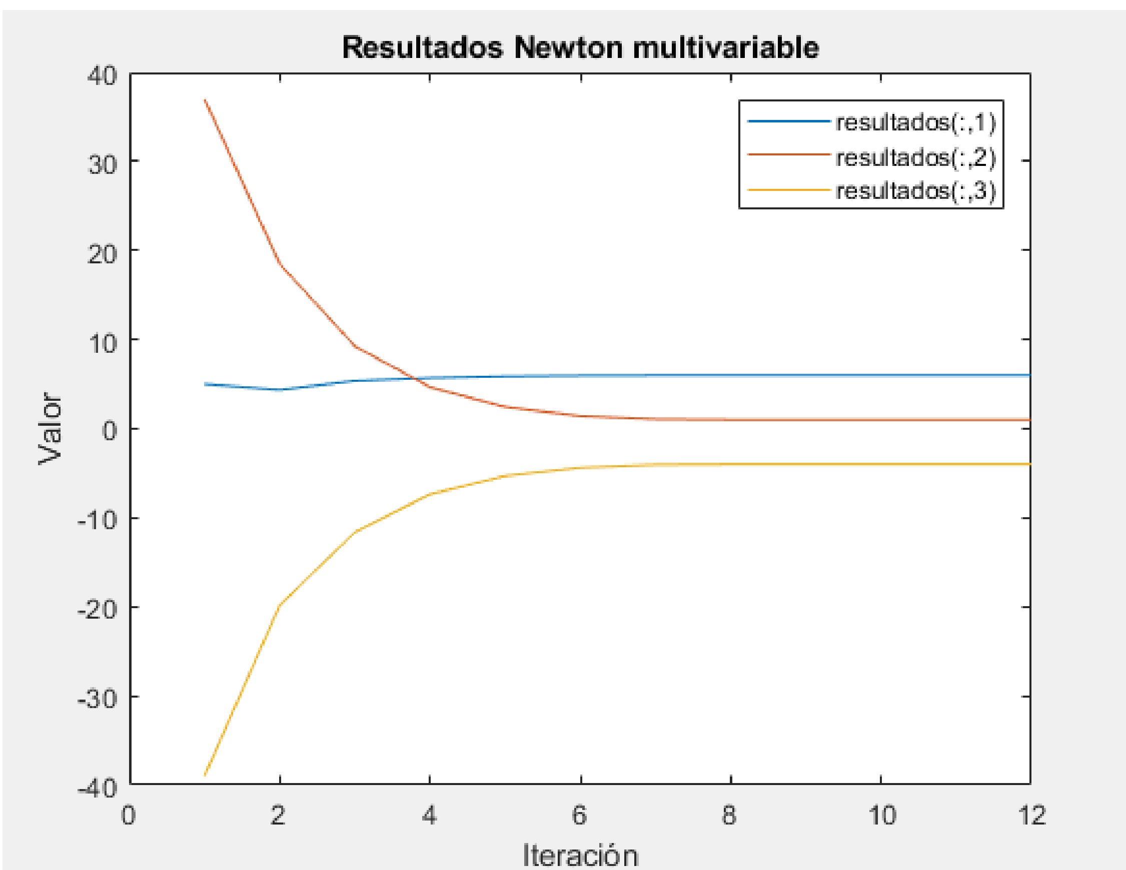

Figura 4-1: Aproximación de la función 1.

La línea azul corresponde a x_1, la roja a x_2 y la amarilla a x_3.

4.2 UTILIZANDO MATRIZ 289X289

Para esta matriz se utilizaron todos los métodos descritos en el capítulo 2, sin contar Newton multivariable, tardando un tiempo total de 135.8079841963097 segundos.

4.2.1 Datos

Los tiempos, errores y costos operacionales de todos los métodos trabajados fueron los siguientes:

Tabla 4.1: Tiempos, errores y costos de métodos con matriz 289x289.

Método	**Tiempo**	**Error**	**Costo**
Gauss.Jacobi	0.508811692412466	-	25085200
Gauss.Seidel	0.375590012087022	-	25027400
LU	0.293802775064269	1.009579602232137e-15	2312
Cholesky	0.952924612484771	1.397193368858244e-15	24597946
QR	1.002947494508918	2.984058425395847e-15	250852
Givens	135.8074070138536	1.454791219327843e-15	749088
Householder	1.590826542182015	7.649509026213863e-15	4624

Los errores de Gauss.Jacobi y Gauss.Seidel se presentan en la sección de gráficos, dado que estos presentan un error por cada iteración.

4.2.2 Gráficos

El siguiente gráfico corresponde a los resultados obtenidos por todos los métodos.

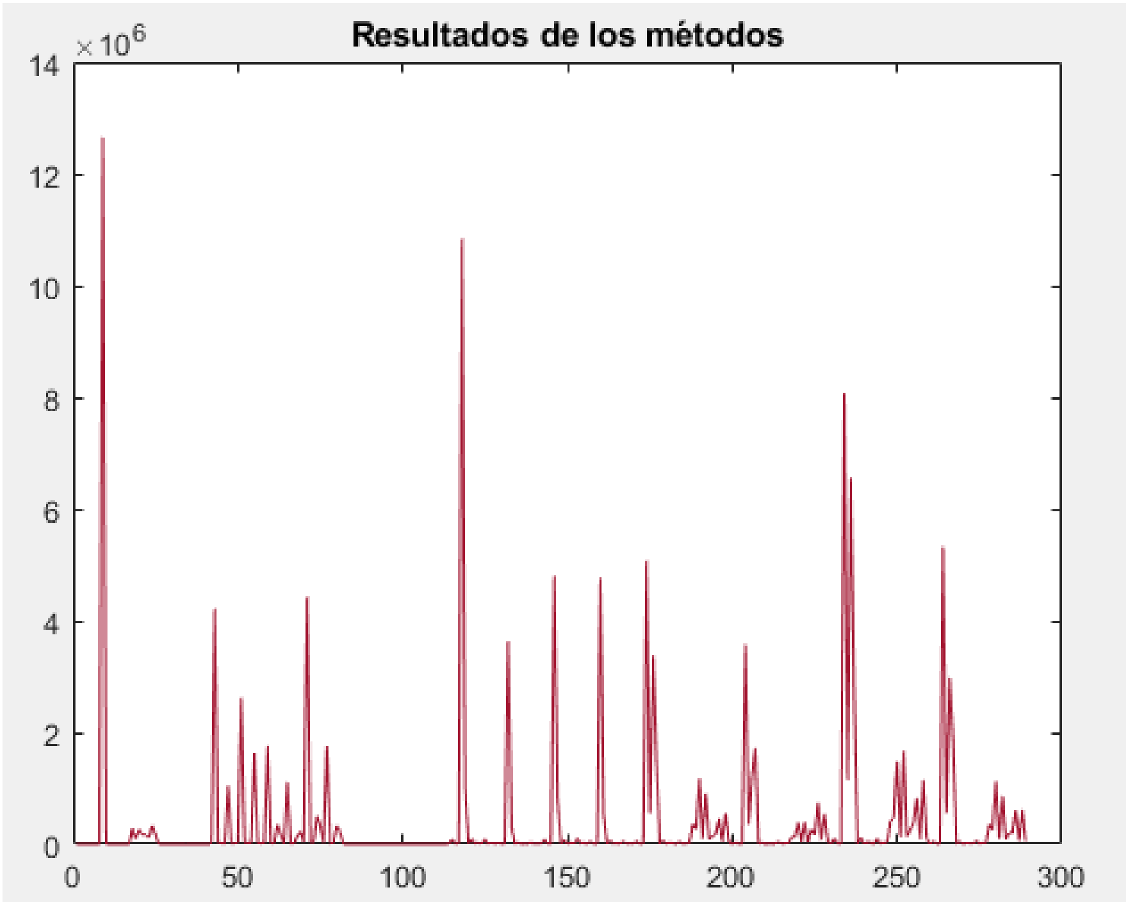

Figura 4-2: Resultados con la matriz 289x289

Los gráficos a continuación corresponden a los errores de Guass.Jacobi y Gauss.Seidel, de los cuales tenemos los errores por iteración, de un total de 100 iteraciones.

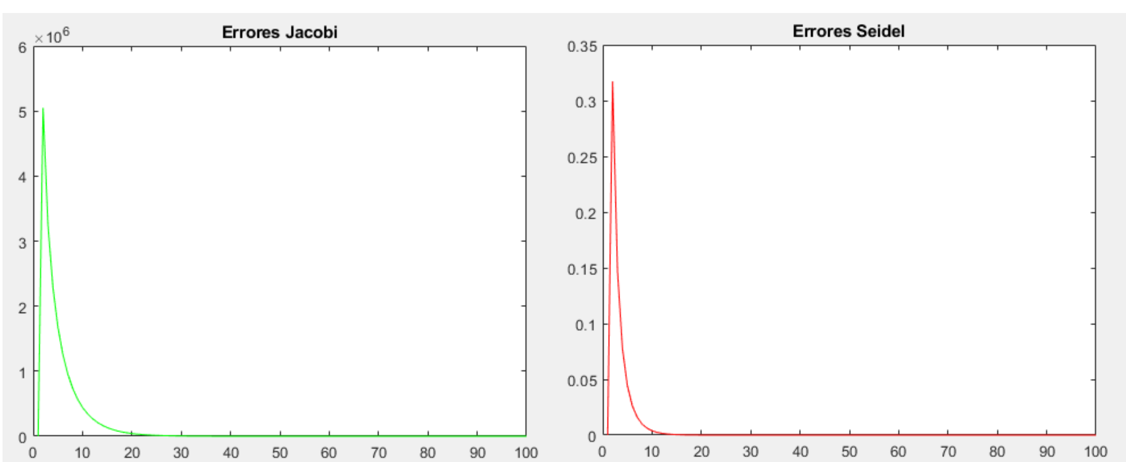

Figura 4-3: Errores Gauss.Jacobi y Gauss.Seidel con matriz 289x289

4.3 UTILIZANDO MATRIZ 1089X1089

Para esta matriz se utilizaron la mayoría de los métodos, exceptuando por quienes no cumplen con los requisitos para ser ocupadas y Givens, el cual tarda mucho tiempo. El tiempo total de ejecución fue de 156.8411214727571 segundos.

4.3.1 Datos

Los tiempos, errores y costos de todos los métodos fueron los siguientes:

Tabla 4.2: Tiempos, errores y costos de métodos con matriz 1089x1089.

Método	Tiempo	Error	Costo
Gauss.Jacobi	10.415334095664972	-	355885200
LU	7.931765683242751	2.645404174808627e-15	8712
QR	12.246252238393259	1.159115526814099e-14	3558852
Householder	156.8355174899567	2.373913887811701e-14	17424

Al igual que en la matriz 289x289 el error de Gauss.Jacobi se presenta en la sección de gráficos.

4.3.2 Gráficos

El siguiente gráfico corresponde a los resultados obtenidos por los métodos presentados con anterioridad.

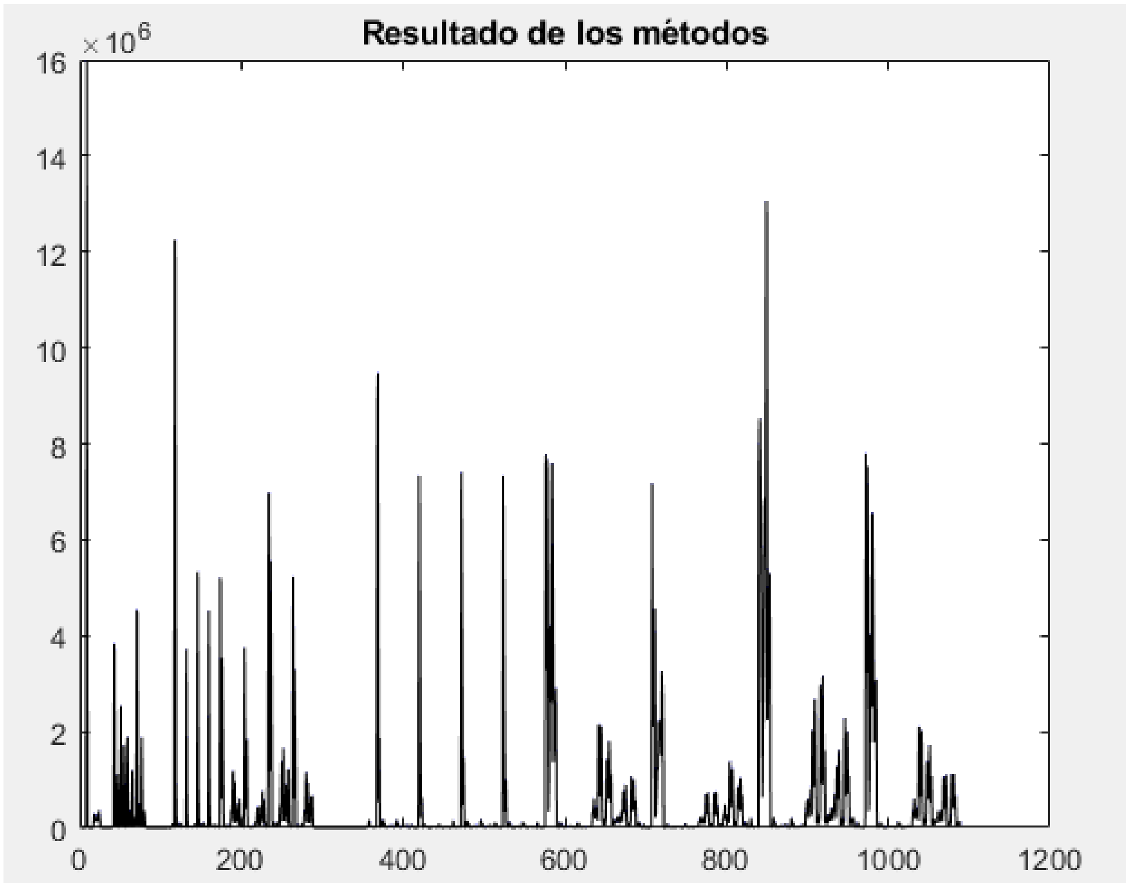

Figura 4-4: Resultados con la matriz 1089x1089

El gráfico a continuación es correspondiente a los errores que tuvo Gauss.Jacobi por iteración, con 100 iteraciones.

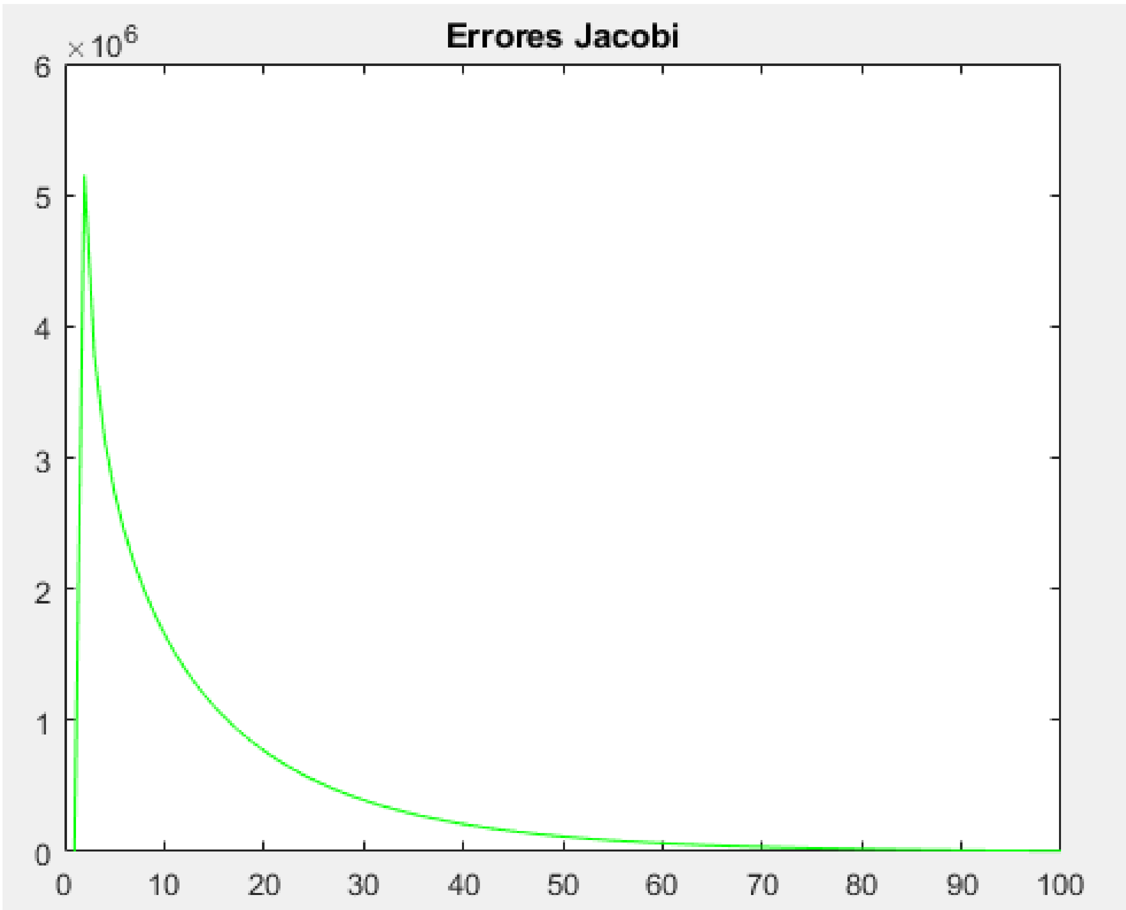

Figura 4-5: Errores Gauss.Jacobi con matriz 1089x1089

4.4 UTILIZANDO MATRIZ 4225X4225

Para esta matriz se utilizaron 4 métodos: Gauss.Jacobi, LU, QR y Householder. El resto de los métodos no se ocuparon debido a que la matriz no cumplía con las características necesarias para trabajarlas, y en el caso de Givens por un tema de tiempo. El tiempo total de ejecución fue de 20756.96298726938 segundos.

4.4.1 Datos

Los tiempos, errores y costos de los métodos mencionados fueron los siguientes:

Tabla 4.3: Tiempos, errores y costos de métodos con matriz 4225x4225.

Método	**Tiempo[s]**	**Error**	**Costo**
Gauss.Jacobi	78.299145939014280	-	5355610000,0
LU	848,601959039366	1,28656685648581e-14	33800
QR	20756,6847647982	8,43707475179689e-14	53556100
Householder	433,560002221790	1,29544410377674e-13	67600

Como se dijo anteriormente el error de Gauss.Jacobi es una matriz, donde se encuentran todos los errores de todas las iteraciones, y por ende se graficará aparte.

4.4.2 Gráficos

El siguiente gráfico corresponde a los resultados obtenidos por los métodos mencionados con anterioridad, donde el color azul corresponde a Gauss.Jacobi, el rojo a QR, el amarillo a LU y el lila a Householder.

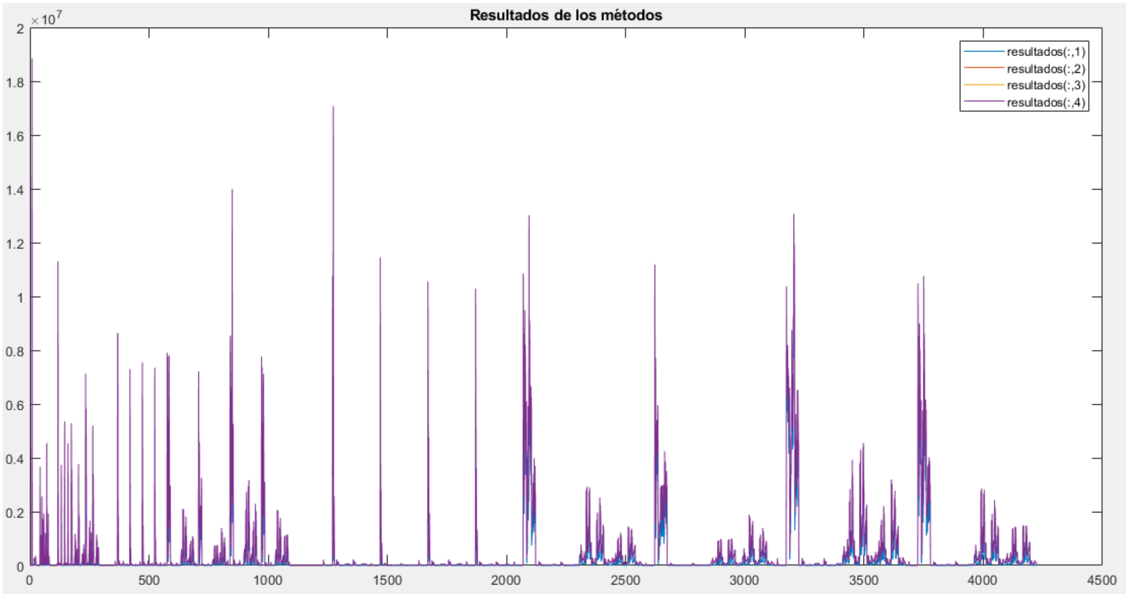

Figura 4-6: Resultados con la matriz 4225x4225

El siguiente gráfico corresponde a los errores de Gauss.Jacobi

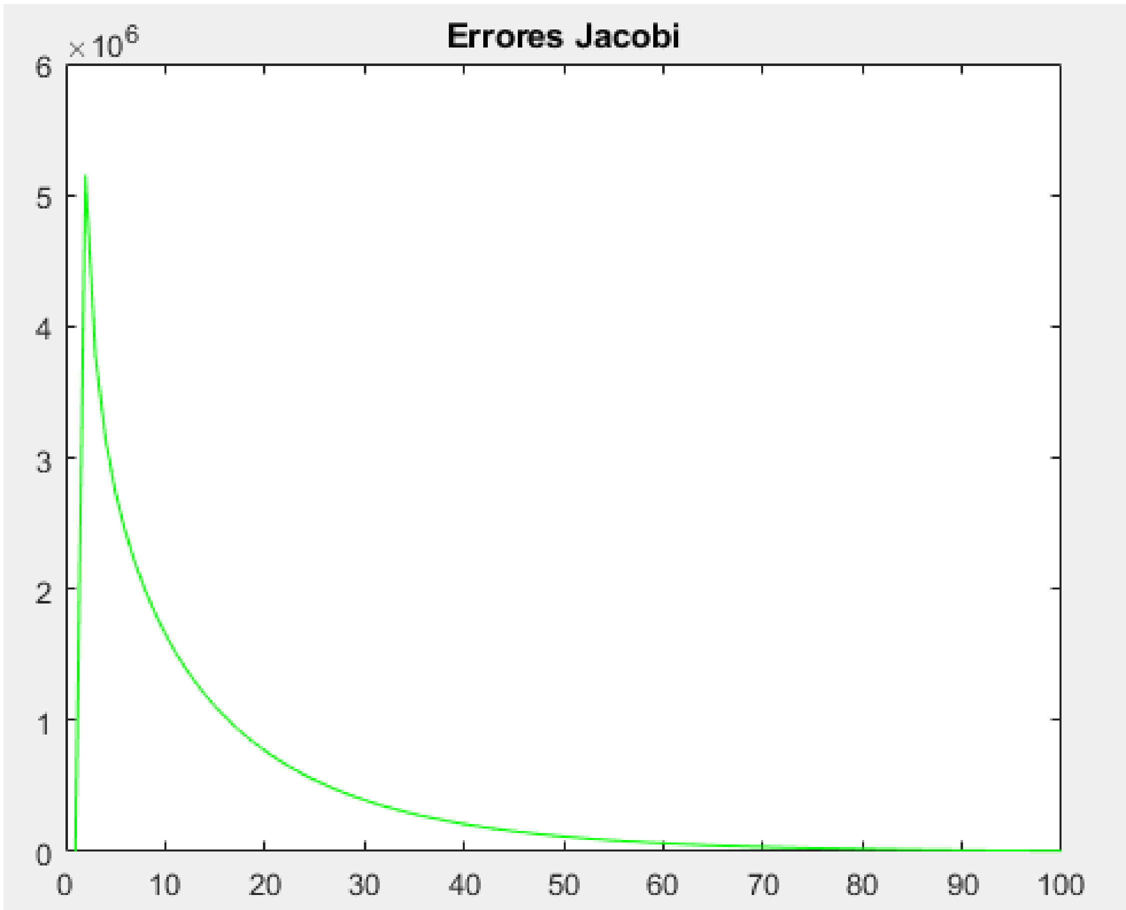

Figura 4-7: Errores Gauss.Jacobi con matriz 4225x4225

CAPÍTULO 5. ANÁLISIS DE RESULTADOS

En este capítulo se realizará un análisis de los resultados mostrados en el capítulo anterios, separando de las misma forma las secciones.

5.1 ANÁLISIS PARA NEWTON MULTIVARIABLE

Recordando el sistema de ecuaciones:

$$\begin{cases} \mathrm{x}_1^2 + x_2 - 37 = 0 \\ \mathrm{x}_1 - x_2^2 - 5 = 0 \\ \mathrm{x}_1 + x_2 + x_3 - 3 = 0 \\ \mathrm{X}_{(0)} = (0, 0, 0)^T \end{cases}$$

A través de la Figura 4-1 se puede apreciar que los resultados de x_1, x_2 y x_3 en cada iteración se acercan más a la solución del sistema, llegando a esta en la iteración 9, entregando los valores: $x_1 = 6$, $x_2 = 1$ y $x_3 = -4$.

5.2 ANÁLISIS PARA LA MATRIZ 289X289

Dado que esta matriz es simétrica, definida positiva, dominante, dispersa, semi positiva y negativa, todos los métodos fueron utilizados.

Con los datos puestos en la Tabla 4.1, podemos apreciar que el método Givens es el que más tiempo consume, pero a su vez uno de los que tiene el error menor. Pese a consumir tanto tiempo, el costo operacional de Givens no es tan alto, comparado con el de Gauss.Jacobi o Gauss.Seidel, debido a que esos son métodos iterativos y no directo como Givens.

Si se comparan los métodos iterativos, el que más rápido ejecuta con un error menor y costo operacional menor es Gauss.Seidel, sin embargo éste método tiene una restricción que Gauss.Jacobi no la tiene, aparte de que la matriz ingresada sea dominante, esta debe ser simétrica y definida positiva. Por lo mismo se puede utizar Gauss.Seidel en una cantidad menor de matrices que Gauss.Jacobi. Tratándose de errores, ambos métodos tienen gráficos similares (Figura 4-3), solo que a distinta escala, dado que Gauss.Seidel tiene errores mucho menores en sus iteraciones.

Con respecto a los métodos directos, quienes tienen mayor tiempo de ejecución son

Givens y Householder, ganando lejos Givens. En términos de menor error se tiene a LU, seguido por Cholesky, Givens, QR y finalmente Householder. Quien realiza un mayor costo operacional viene siendo Cholesky, y el de menos costo operacional es LU, de aquí podemos decir que para matrices de este tamaño conviene LU, teniendo en cuenta que es el método con menor tiempo, menor error y menor costo operacional.

Pese a lo anterior mencionado, todos los métodos entregan un resultado muy cercano, tal y como se aprecia en la Figura 4-2, esto puede ser dado ya que el uso de aproximaciones es menor al trabajar con una matriz de pequeña.

5.3 ANÁLISIS PARA LA MATRIZ 1089X1089

Esta matriz es simétrica, dispersa, dominante y semi definida positiva, por lo tanto solo se trabaja con 5 métodos, LU, Givens, Householder, QR (con gram-schmidt) y Gauss.Jacobi. De esos métodos no se utilizó Givens ya que al recorrer la matriz, luego alrededor de 9 horas recién alcanzó a ejecutar la columna 205, en la fila 252.

A partir de los resultados presentados en el capítulo anterior en la Figura 4-4, podemos deducir que los métodos trabajados se comportan de forma similar, sin destacar mayor diferencia en los resultados.

Viendo la Tabla 4.2 se distingue que el método en consumir mayor tiempo es Householder, y a la vez es quien tiene mayor error (entre los métodos directos), pero su costo operacional es mucho menor que el de QR, y mayor que LU.

Con respecto a todos los métodos utilizados, LU sigue siendo el que menos tiempo tarda, con menor error y menor costo operacional, de esta forma podemos decir, al igual que en la matriz anterior, que LU es el que más conviene, ya que entrega un resultado muy similar que el resto, pero con menor tiempo, error y costo.

5.4 ANÁLISIS PARA LA MATRIZ 4225X4225

La matriz de 4225x4225 es dispersa, dominante, semi definida positiva y simétrica, por lo que solo se trabaja con 5 métodos, LU, Givens, Householder, QR y Gauss.Jacobi. De esos métodos Givens no se utilizó por motivos de tiempo, pero de todas formas se puede realiza un análisis del resto de los métodos.

En general los resultados de los métodos utilizados siguen un comportamiento similar, lo que se demuestra en la Figura 4-6, sin embargo se aprecia cada vez una una mayor diferencia entre dos métodos, Gauss.Jacobi (color azul) y Housholder(color lila). Estas pequeñas diferencias, y el tiempo que invirtió cada método demuestra que Gauss.Jacobi aproxima más que Householder, por eso es más rápido, pero a la vez menos certero. Estas pequeñas diferencias empiezan a crecer o hacerse notar debido a la matriz, que es mucho más grande que las trabajadas anteriormente.

Como se aprecia en la Tabla 4.3, quien tiene mayor costo operacional es Gaus.Jacobi, pese a eso es quien se demora menos en ejecutar.

Comparando los métodos directos, esta vez es Householder quien tarda menos, y QR es quien más se demora. Aunque el error de Householder es levemente menor con respecto a los otros, sigue siendo más recomendable para matrices grandes como esta, ya que tarda aproximadamente la mitad menos que LU, aunque tenga casi el doble de costos operacionales, pese a lo recién mencionado el resultado es muy cercano a los que el resto de los métodos entregaron.

CAPÍTULO 6. CONCLUSIONES

Luego del análisis de los resultados con respecto a las matrices, presentado en el capítulo anterior, se puede apreciar que las diferencias entre los distintos métodos, al calcular los resultados, no son tan notorias, pero a medida que la matriz es más grande estos resultados se van alejando, esto debido a la aproximación que se ocupa en cada método.

A la hora utilizar un método se debe tener en cuenta lo siguiente:

- Las características de las matrices. Para efecto de este laboratorio se analizaron 6 tipos, y dependiendo de eso existían métodos como Cholesky que no podían utilizarse para ciertas matrices. Otro ejemplo es el caso de Gauss.Seidel, el cual es mucho mejor que Gauss.Jacobi, tanto en tiempo, como en error y costo operacional, ya que realiza la mitad de las iteraciones que Gauss.Jacobi, pese a eso no se pudo utilizar en las siguientes matrices.

- El tiempo de cada método. Se logró apreciar anteriormente que algunos métodos tardan mucho más que el resto, por este motivo depende si se requiere una aproximación más certera, o unos resultados aproximados pero con respuesta más rápida.

- Cantidad de operaciones. El método de Gauss.Jacobi en particular realizaba más operaciones, pese a eso su tiempo es menor a otros. Pero si se desea ocupar un método con menor cantidad de operaciones, se debe considerar buscar otro método.

- Tamaño de la matriz. Dependiendo del tamaño que tenga la matriz afecta el tiempo y error del método. Para matrices pequeñas y medianas, LU es el que mejor trabaja, pero si se ocupa una matriz grande como la de 4225x4225, en ese caso es mejor Householder.

Además, en el caso de la computadora, es importante mencionar que los métodos numéricos tienen ciertas limitaciones como los siguientes:

- Tamaño de los registros del procesador. Los registros del procesador pueden almacenar una serie de datos de un cierto tamaño; pero esto es limitado, lo que provoca que la precisión no sea 100 % certera, ya que cuando un número sobrepasa el tamaño del registro, este número se ve aproximado o truncado, perdiendo decimales de precisión.

- Lenguaje de programación. En este laboratorio se utilizó MATLAB, que está diseñado justamente para realizar cálculos matemáticos con distintos tipos de complejidades. Es por este motivo que los cálculos se realizan con muchos decimales, aprovechando así al máximo MATLAB para las iteraciones. Si se hubiese desarrollado el programa en otro lenguaje como python, las aproximaciones no habrían sido tan cercanas a las raíces de las funciones, en el caso de Newton mutivariable.

Finalmente, se logró implementar todos los métodos solicitados, logrando utilizar la mayoría de estos, exeptuando a Givens. También se logró tener un manejo básico en MATLAB, permitiendo a futuro poder utilizar de mejor manera esta herramienta, ya sea en los siguientes laboraorios, como en otras circunstancias. Por otra parte, al realizar un análisis profundo sobre estos métodos, se puede lograr comprender de mejor manera los conceptos vistos en cátedra.

CAPÍTULO 7. BIBLIOGRAFÍA

Argos, C. G. (2002/2003). *Apuntes de Métodos Numéricos*. Universidad de Málaga.

Plaza, S. (2007). *Métodos Numéricos*. Universidad de Santiago de Chile.