

LABORATORIO 2 DE ALGORITMOS NUMÉRICOS

DANY RUBIANO JIMENEZ

Profesor: Oscar Rojas D.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS.....	v
ÍNDICE DE CUADROS	vi
CAPÍTULO 1. INTRODUCCIÓN.....	7
1.1 MOTIVACIÓN Y ANTECEDENTES	7
1.2 OBJETIVOS	7
1.3 ORGANIZACIÓN DEL DOCUMENTO	7
CAPÍTULO 2. DESCRIPCIÓN DE LOS MÉTODOS	9
2.1 MÉTODOS DIRECTOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES	9
2.1.1 Factorización LU	9
2.1.2 Factorización QR	9
2.2 MÉTODOS ITERATIVOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES	10
2.2.1 Método de Gauss-Seidel	10
2.2.2 Método SOR	10
2.3 INTERPOLACIÓN POLINOMIAL	10
2.3.1 Polinomio de Lagrange	11
2.3.2 Fórmula de Newton	11
2.3.3 Aproximación por mínimos cuadrados	11
2.4 MÉTODOS DE AJUSTE DE FUNCIONES	11
2.4.1 Splines	11
2.4.2 Splines Cúbicos	12
CAPÍTULO 3. RESULTADOS.....	13
3.1 MÉTODOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES	13
3.1.1 Espectro 289 x 289	13
3.1.2 Espectro 1089 x 1089	16
3.1.3 Espectro 4225 x 4225	18
3.2 MÉTODOS DE INTERPOLACIÓN	21

3.2.1	$f_1(x)$	21
3.2.1.1	Diferencia entre los puntos de 5	21
3.2.1.2	Diferencia entre los puntos de 2	22
3.2.1.3	Diferencia entre los puntos de 1	23
3.2.1.4	Diferencia entre los puntos de 0.5	24
3.2.2	$f_2(x)$	25
3.2.2.5	Diferencia entre los puntos de 5	25
3.2.2.6	Diferencia entre los puntos de 2	26
3.2.2.7	Diferencia entre los puntos de 1	27
3.2.2.8	Diferencia entre los puntos de 0.5	28
CAPÍTULO 4. ANÁLISIS DE LOS RESULTADOS		29
4.1	MÉTODOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES	29
4.2	MÉTODOS DE INTERPOLACIÓN	29
CAPÍTULO 5. CONCLUSIONES		31
CAPÍTULO 6. BIBLIOGRAFÍA.....		33

ÍNDICE DE FIGURAS

Figura 3-1: Espectro vector solución a matriz 289 x 289	13
Figura 3-2: Error de aproximación de matriz 289 x 289	15
Figura 3-3: Error con escala logarítmica de aproximación de matriz 289 x 289	16
Figura 3-4: Espectro vector solución a matriz 1089 x 1089	17
Figura 3-5: Error de aproximación de matriz 1089 x 1089	18
Figura 3-6: Error con escala logarítmica de aproximación de matriz 1089 x 1089	18
Figura 3-7: Espectro vector solución a matriz 4225 x 4225	19
Figura 3-8: Error de aproximación de matriz 4225 x 4225	20
Figura 3-9: Error con escala logarítmica de aproximación de matriz 4225 x 4225	20
Figura 3-10: Interpolación para $f_1(x)$, con diferencia de 5 entre los puntos	21
Figura 3-11: Interpolación para $f_1(x)$, con diferencia de 2 entre los puntos	22
Figura 3-12: Interpolación para $f_1(x)$, con diferencia de 1 entre los puntos	23
Figura 3-13: Interpolación para $f_1(x)$, con diferencia de 0.5 entre los puntos	24
Figura 3-14: Interpolación para $f_2(x)$, con diferencia de 5 entre los puntos	25
Figura 3-15: Interpolación para $f_2(x)$, con diferencia de 2 entre los puntos	26
Figura 3-16: Interpolación para $f_2(x)$, con diferencia de 1 entre los puntos	27
Figura 3-17: Interpolación para $f_2(x)$, con diferencia de 0.5 entre los puntos	28

ÍNDICE DE CUADROS

Tabla 3.1: Cantidad de operaciones por método para la matriz de 289 x 289	14
Tabla 3.2: Error RMSE en comparación con función 'linsolve' de MATLAB para la matriz de 289 x 289	15
Tabla 3.3: Cantidad de operaciones por método para la matriz de 1089 x 1089	17
Tabla 3.4: Error RMSE en comparación con función 'linsolve' de MATLAB para la matriz de 1089 x 1089	17
Tabla 3.5: Cantidad de operaciones por método para la matriz de 4225 x 4225	19
Tabla 3.6: Error RMSE en comparación con función 'linsolve' de MATLAB para la matriz de 4225 x 4225	19
Tabla 3.7: Error RMSE en comparación con valores reales de la función $f1_x$	21
Tabla 3.8: Error RMSE en comparación con valores reales de la función $f1_x$	22
Tabla 3.9: Error RMSE en comparación con valores reales de la función $f1_x$	23
Tabla 3.10: Error RMSE en comparación con valores reales de la función $f1_x$	24
Tabla 3.11: Error RMSE en comparación con valores reales de la función $f2_x$	25
Tabla 3.12: Error RMSE en comparación con valores reales de la función $f2_x$	26
Tabla 3.13: Error RMSE en comparación con valores reales de la función $f2_x$	27
Tabla 3.14: Error RMSE en comparación con valores reales de la función $f2_x$	28

CAPÍTULO 1. INTRODUCCIÓN

1.1 MOTIVACIÓN Y ANTECEDENTES

Los métodos numéricos son técnicas mediante las cuales es posible formular problemas matemáticos de tal forma que puedan resolverse usando operaciones aritméticas.

Estos métodos nos vuelven aptos para entender esquemas numéricos a fin de resolver problemas matemáticos, de ingeniería y científicos en una computadora, reducir esquemas numéricos básicos, escribir programas y resolverlos en una computadora y usar correctamente el software existente para dichos métodos y no solo aumenta nuestra habilidad para el uso de computadoras sino que también amplía la pericia matemática y la comprensión de los principios científicos básicos. El análisis numérico trata de diseñar métodos para “aproximar” de una manera eficiente las soluciones de problemas expresados matemáticamente.

El objetivo principal del análisis numérico es encontrar soluciones “aproximadas” a problemas complejos utilizando sólo las operaciones más simples de la aritmética. Se requiere de una secuencia de operaciones algebraicas y lógicas que producen la aproximación al problema matemático.

1.2 OBJETIVOS

El objetivo principal de este laboratorio es analizar las características de los diferentes métodos presentados, considerando la rapidez con que convergen a la solución y el tamaño del error de cada uno de ellos, tomando en cuenta todas las iteraciones realizadas. Para esto, es necesario programar los algoritmos revisados en las clases de cátedra, consiguiendo manejo básico de la programación en MATLAB; y utilizando los gráficos que puede realizar este programa para de esta forma, consolidar los conocimientos obtenidos en las clases teóricas.

1.3 ORGANIZACIÓN DEL DOCUMENTO

Este informe comienza con una breve descripción de los métodos, en donde se encuentra información relevante para el entendimiento de cada uno de ellos. Seguido se muestran mediante gráficos, información del cómo actúan cada uno de los métodos aplicados a distintas funciones predeterminadas, las cuales se informan en su respectivo momento, en específico las mencionadas gráficas deben proporcionar información sobre errores, cantidad de iteraciones, además las raíces respectivas. Finalmente se realizan conclusiones correspondientes con los datos obtenidos de los datos del análisis.

CAPÍTULO 2. DESCRIPCIÓN DE LOS MÉTODOS

2.1 MÉTODOS DIRECTOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES

2.1.1 Factorización LU

Este método de resolución de ecuaciones lineales consta en separar la matriz A en dos matrices, L y U , donde la matriz L (Lower) es una matriz triangular inferior con unos en su diagonal y U (Upper) es una matriz triangular superior.

$$Ax = B$$

Para descomponer la matriz A , se debe resolver la siguiente ecuación:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ L_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & L_{n3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Una vez que se resuelve $A = LU$, finalmente se debe encontrar la matriz X , que es lo que se busca en un comienzo. Para esto, se debe resolver un sistema más sencillo.

$$Ly = B$$

$$Ux = y$$

Donde y será una matriz auxiliar para realizar los cálculos.

2.1.2 Factorización QR

Dada una matriz A (no necesariamente cuadrada) con columnas linealmente independientes, encontraremos Q , R tal que:

1. $A = Q * R$
2. Las columnas de Q son ortonormales
3. Q es del mismo tamaño de A
4. R es triangular superior invertible

Para esto, se debe utilizar el proceso de Gram-Schmidt para todas las columnas (v_1, v_2, \dots, v_n) de la matriz A , obteniendo las columnas que corresponderán a las columnas de la matriz Q .

$$u_1 = v_1$$

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

$$Q = (u_1 u_2 \cdots u_n)$$

Donde los vectores obtenidos $(u_1 u_2 \cdots u_n)$ son ortonormales.

Una vez obtenida la matriz Q a partir del proceso de Gram-Schmidt sólo basta calcular la matriz R.

$$R = Q^t A$$

2.2 MÉTODOS ITERATIVOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES

2.2.1 Método de Gauss-Seidel

Sea A una matriz y el sistema de ecuaciones $Ax = B$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

El método Gauss-Seidel utiliza los valores obtenidos de la ecuación $(i-1)$ para aproximar los resultados de la ecuación i -ésima. Es decir, para obtener, por ejemplo, X_2^{k+1} , utiliza lo computado en X_1^{k+1} (2.15).

$$X_i^{(k)} = \frac{1}{a_{ii}} \left[- \sum_{j=1}^{i-1} (a_{ij} x_j^{(k)}) - \sum_{j=i+1}^n (a_{ij} x_j^{(k-1)}) + b_i \right]$$

2.2.2 Método SOR

El método SOR es parecido al método de Gauss-Seidel, pero utiliza un factor de escala para reducir más rápidamente el error de la aproximación. La técnica SOR pertenece a una clase de métodos de relajación que calculan las aproximaciones $x_{(k)}$ mediante la siguiente expresión.

$$X_i^{(k)} = (1-w)x_i^{(k)} + \frac{w}{a_{ii}} \left[- \sum_{j=1}^{i-1} (a_{ij} x_j^{(k)}) - \sum_{j=i+1}^n (a_{ij} x_j^{(k-1)}) + b_i \right]$$

Donde w es el factor de escala.

2.3 INTERPOLACIÓN POLINOMIAL

Dado $n+1$ puntos distintos, x_i ($i = 0, 1, \dots, n$) y los valores de la función en esos puntos $f(x_i) = y_i$ ($i = 0, 1, \dots, n$) el polinomio de menor grado (n) que coincida con f en esos puntos se denomina "polinomio de interpolación de f en los puntos x_i ($i = 0, 1, \dots, n$)". Llamemos a éste polinomio $P(x)$ de grado n tal que:

$$P(x_i) \approx f(x_i) = y_i; i = 0, 1, \dots, n$$

Para encontrar dicho polinomio $P(x)$ existen diversos métodos que serán explicados a continuación.

2.3.1 Polinomio de Lagrange

Una forma conocida de encontrar de forma directa el polinomio de interpolación es a través de la fórmula de Lagrange.

$$P(x) = \sum_{i=0}^n y_i L_i(x)$$

Donde:

$$L_i = \prod_{k=0}^n \frac{(x - x_k)}{(x_i - x_k)}$$

Con k distinto de i .

2.3.2 Fórmula de Newton

El polinomio de interpolación de grado n -ésimo de Newton es formado según la ecuación:

$$f_n(x) = b_0 + b_1(x - x_0) + \dots + b_n(x - x_0) \dots (x - x_{n-1})$$

En donde cada uno de los b_i , serán llamados las diferencias divididas, dados por la fórmula:

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0}$$

2.3.3 Aproximación por mínimos cuadrados

Sea $P(x)$ el polinomio de interpolación de grado n , de la forma:

$$P_n(x) = a_0 + a_1x + \dots + a_nx^n$$

Los coeficientes del polinomio aproximado estarán dados por la solución al sistema.

$$\begin{bmatrix} n & \sum x_i & \dots & \sum x_i^{n-1} \\ \sum x_i & \sum x_i^2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^{m-1} & \sum x_i^m & \dots & \sum x_i^{2m-2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \vdots \\ \sum y_i x_i^{n-1} \end{bmatrix}$$

2.4 MÉTODOS DE AJUSTE DE FUNCIONES

2.4.1 Splines

Una función spline está formada por varios polinomios, cada uno definido sobre un subintervalo, que se unen entre sí obedeciendo a ciertas condiciones de continuidad.

Sean

$$t_0 < t_1 < \dots < t_n$$

nodos del polinomio que se quiere aproximar. Se dice que una función spline de grado k con nodos $t_0 < t_1 < \dots < t_n$ es una función S que satisface las siguientes condiciones:

- En cada intervalo $[t_{i-1}, t_i)$, S es un polinomio de grado menor o igual a k .
- S tiene una derivada de orden $(k - 1)$ continua en $[t_0, t_n]$

Un spline de grado 1 se puede definir por:

$$S(x) = \begin{cases} S_0(x) = a_0x + b_0 & , x \in [t_0, t_1) \\ S_1(x) = a_1x + b_1 & , x \in [t_1, t_2) \\ \vdots & \\ S_{n-1}(x) = a_{n-1}x + b_{n-1} & , x \in [t_{n-1}, t_n] \end{cases}$$

2.4.2 Splines Cúbicos

El spline cúbico ($k=3$) es el spline más utilizado, debido a que proporciona un excelente ajuste a los puntos tabulados y su cálculo no es excesivamente complejo. Sobre cada intervalo $[t_0, t_1], \dots, [t_{n-1}, t_n]$ S está definido por un polinomio cúbico diferente. Sea S_i el polinomio cúbico que represente a S en el intervalo $[t_i, t_{i+1}]$, entonces se tendrá los siguientes polinomios para cada intervalo conocido.

$$S_i(x) = y_i + (x - t_i)[C_i + (x - t_i)[B_i + (x - t_i)A_i]]$$

En donde:

$$\begin{aligned} A_i &= \frac{1}{bh_i}(z_{i+1} - z_i) \\ B_i &= \frac{z_i}{2} \\ C_i &= -\frac{h_i z_{i+1}}{b} - \frac{h_i z_i}{3} + \frac{1}{h_i}(y_{i+1} - y_i) \\ h_i &= t_{i+1} - t_i \end{aligned}$$

CAPÍTULO 3. RESULTADOS

3.1 MÉTODOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES

En esta sección se presentan mediante gráficos y tablas los resultados y errores estimados obtenidos al utilizar los métodos de LU, QR, Gauss-Seidel y SOR. Es necesario mencionar que para estos dos últimos métodos en estudio se utilizó un nivel de tolerancia de 10^{-5} ya que estos funcionan de forma iterativa.

Se incluyen por lo tanto, los resultados obtenidos en la aplicación de estos métodos a cada una de las matrices dadas, que corresponden a los espectros 289×289 , 1089×1089 , y 4225×4225 .

3.1.1 Espectro 289×289

Teniendo en cuenta que la tolerancia es de 10^{-5} y utilizando un $w = 1,1$ para el método SOR, se obtienen los siguientes resultados:

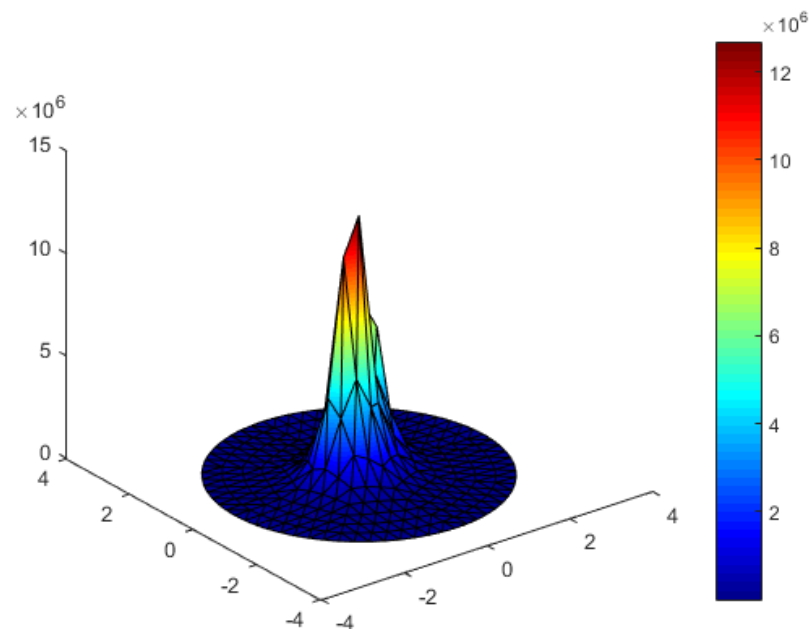


Figura 3-1: Espectro vector solución a matriz 289×289

Tabla 3.1: Cantidad de operaciones por método para la matriz de 289×289

Método	Operaciones	Iteraciones
LU	16049904	-
QR	125137	-
Gauss-Seidel	17935340	108
SOR	16147008	97

Tabla 3.2: Error RMSE en comparación con función 'linsolve' de MATLAB para la matriz de 289×289

Método	Error RMSE
LU	4.595275228525527e-10
QR	6.612513628651792e-09
Gauss-Seidel	3.217744101380726e-05
SOR	3.292750056076536e-05

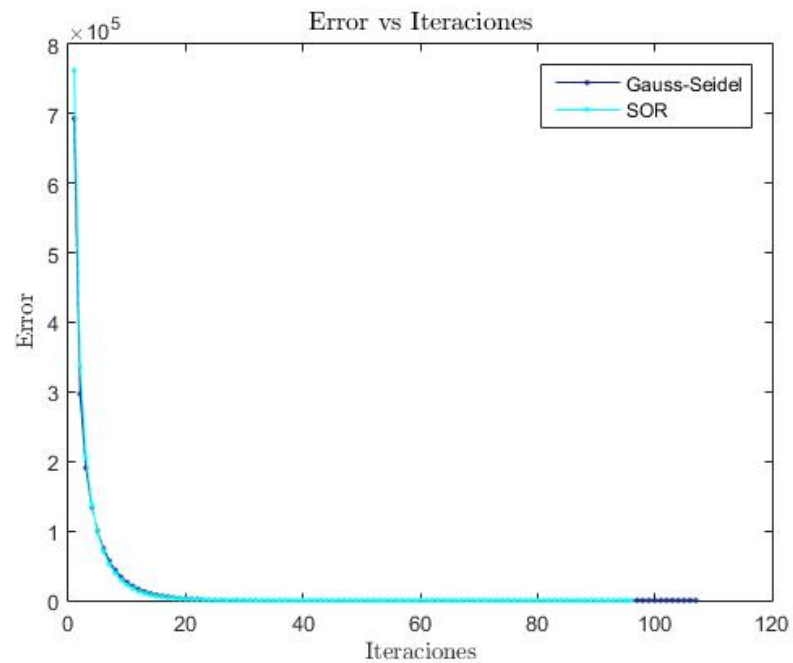


Figura 3-2: Error de aproximación de matriz 289×289

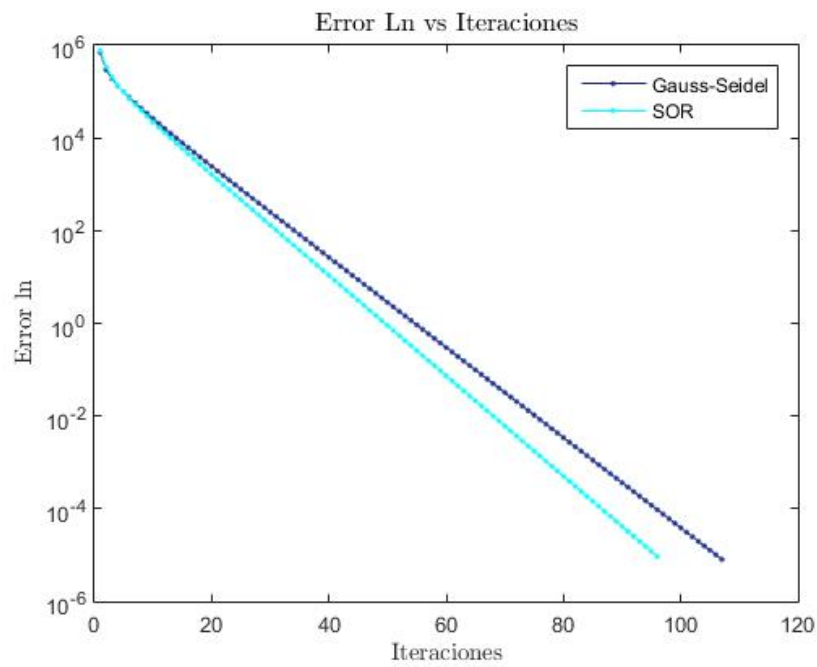


Figura 3-3: Error con escala logarítmica de aproximación de matriz 289×289

3.1.2 Espectro 1089×1089

Teniendo en cuenta que la tolerancia es de 10^{-5} y utilizando un $w = 1,1$ para el método SOR, se obtienen los siguientes resultados:

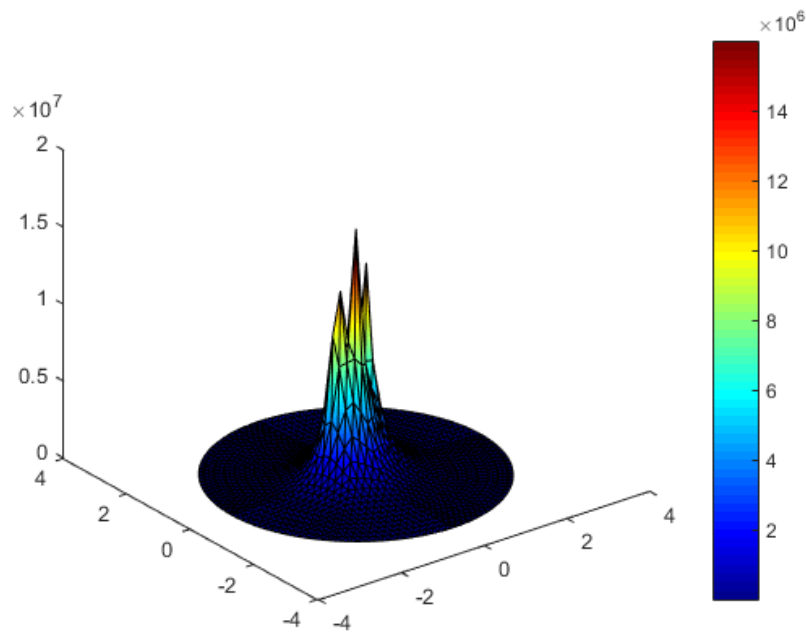


Figura 3-4: Espectro vector solución a matriz 1089×1089

para un $w = 1,1$ y una tolerancia de 10^{-5}

Tabla 3.3: Cantidad de operaciones por método para la matriz de 1089×1089

Método	Operaciones	Iteraciones
LU	860385504	-
QR	1778337	-
Gauss-Seidel	951982020	402
SOR	867312270	366

Tabla 3.4: Error RMSE en comparación con función 'linsolve' de MATLAB para la matriz de 1089×1089

Método	Error RMSE
LU	4.624099558825251e-09
QR	1.027457696207764e-08
Gauss-Seidel	1.671401642225675e-04
SOR	1.524596363508863e-04

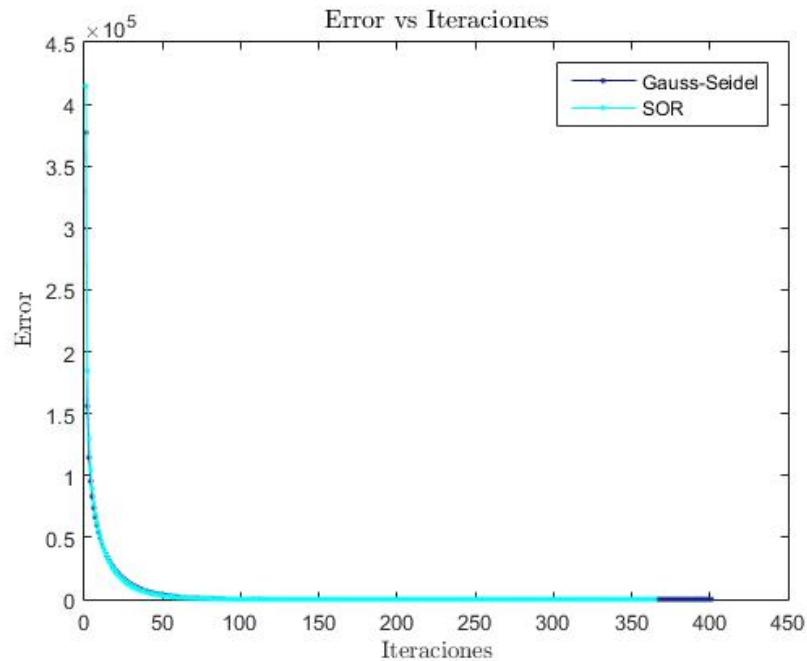


Figura 3-5: Error de aproximación de matriz 1089×1089

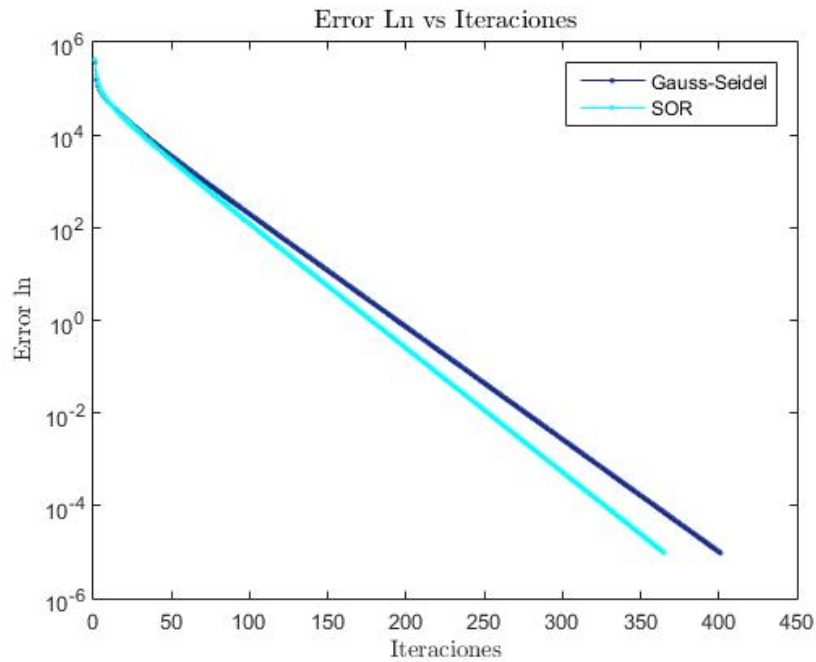


Figura 3-6: Error con escala logarítmica de aproximación de matriz 1089×1089

3.1.3 Espectro 4225×4225

Teniendo en cuenta que la tolerancia es de 10^{-5} y utilizando un $w = 1$ para el método SOR, se obtienen los siguientes resultados:

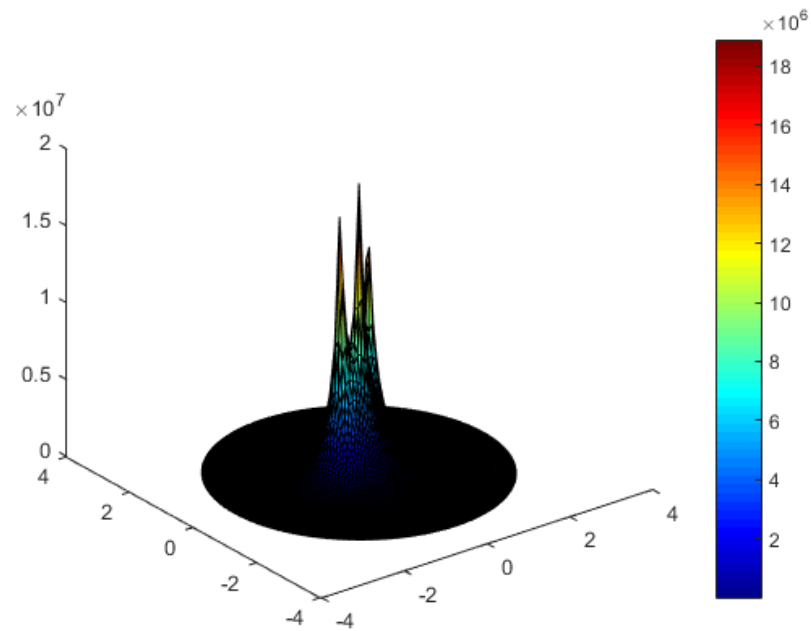


Figura 3-7: Espectro vector solución a matriz 4225×4225

para un $w = 1$ y una tolerancia de 10^{-5} :

Tabla 3.5: Cantidad de operaciones por método para la matriz de 4225×4225

Método	Operaciones	Iteraciones
LU	5.027033440000000e+10	-
QR	26773825	-
Gauss-Seidel	5.367167910000000e+10	1504
SOR	5.367167910000000e+10	1504

Tabla 3.6: Error RMSE en comparación con función 'linsolve' de MATLAB para la matriz de 4225×4225

Método	Error RMSE
LU	2,05742679940346e-08
QR	2.580013047818979e-08
Gauss-Seidel	7.103038629239756e-04
SOR	7.103038629239756e-04

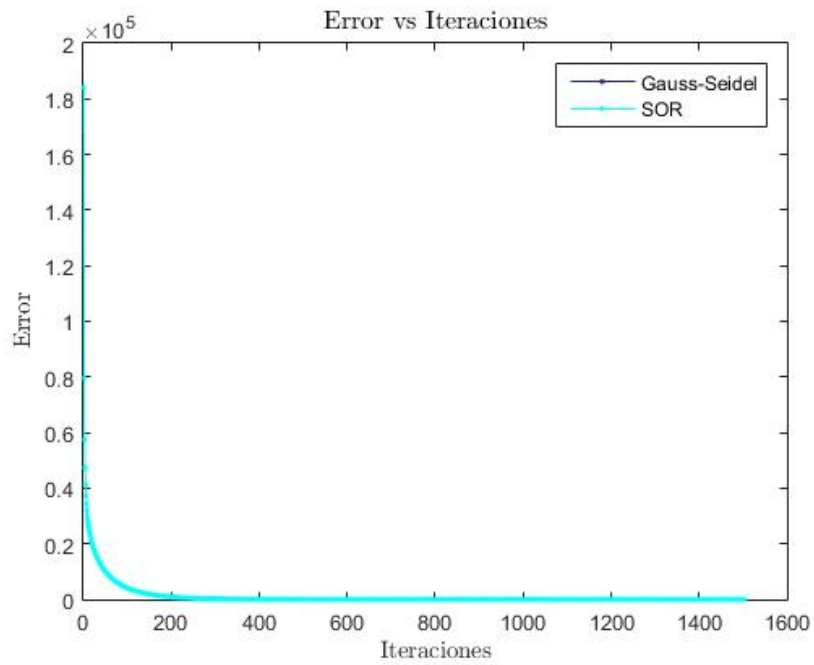


Figura 3-8: Error de aproximación de matriz 4225 x 4225

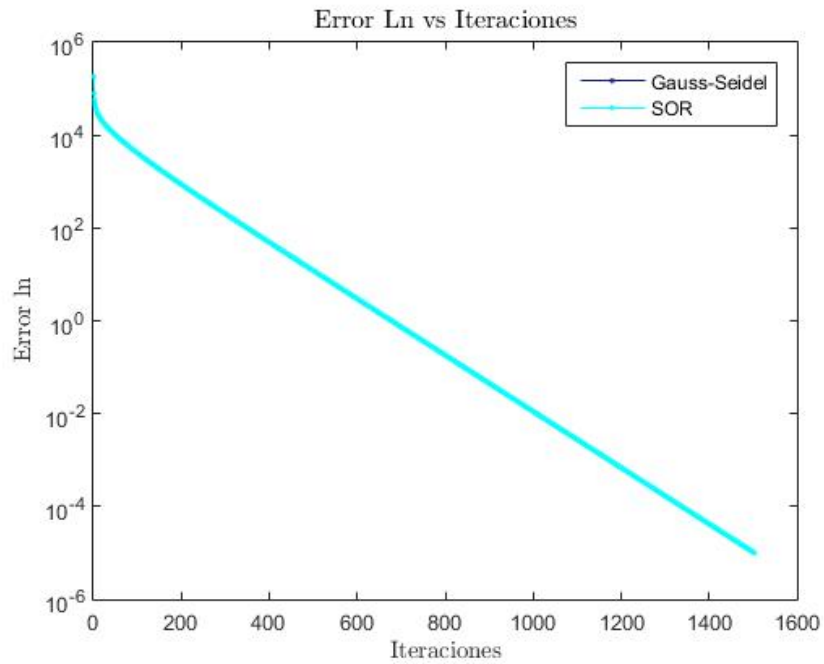


Figura 3-9: Error con escala logarítmica de aproximación de matriz 4225 x 4225

3.2 MÉTODOS DE INTERPOLACIÓN

En esta sección se presentan mediante gráficos y tablas los resultados y errores estimados obtenidos al utilizar los métodos de interpolación de Lagrange, Mínimos Cuadrados, y Spline sobre las funciones dadas, variando la diferencia de unidades entre los puntos de los rangos dados.

3.2.1 $f_1(x)$

Recordar que $f_1(x) = \cos(x) - x^2$

3.2.1.1 Diferencia entre los puntos de 5

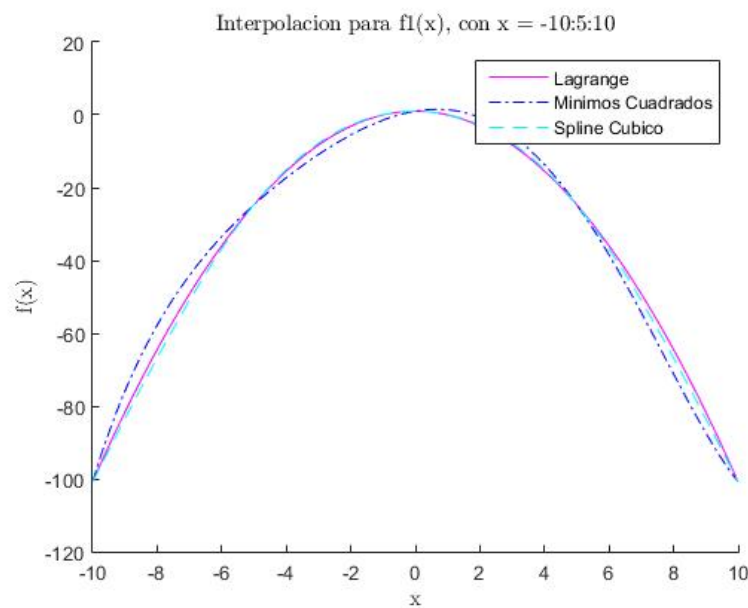


Figura 3-10: Interpolación para $f_1(x)$, con diferencia de 5 entre los puntos

Tabla 3.7: Error RMSE en comparación con valores reales de la función $f_1(x)$

Método	Error RMSE
Lagrange	0.904595451934607
Mínimos Cuadrados	3.688191810723940
Spline Cúbico	1.761668335422684

3.2.1.2 Diferencia entre los puntos de 2

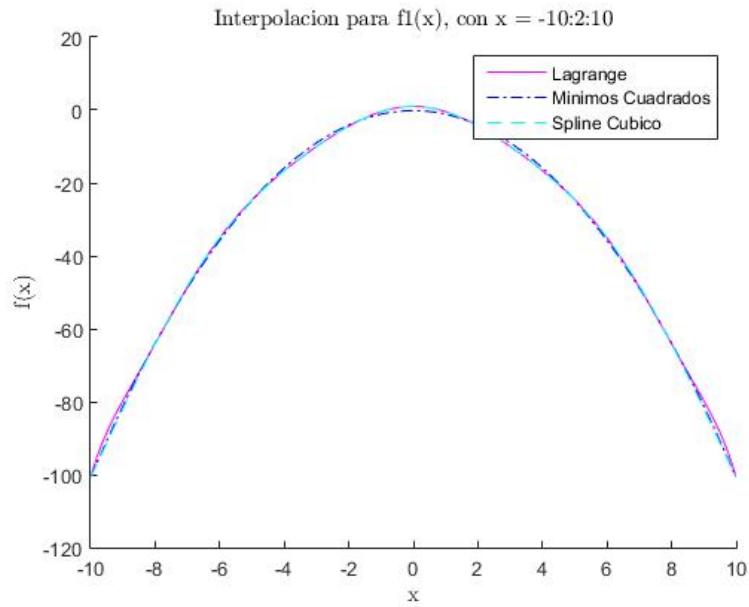


Figura 3-11: Interpolación para $f_1(x)$, con diferencia de 2 entre los puntos

Tabla 3.8: Error RMSE en comparación con valores reales de la función f_{1x}

Método	Error RMSE
Lagrange	0.832711627861061
Mínimos Cuadrados	0.638839829074685
Spline Cúbico	0.047463607510147

3.2.1.3 Diferencia entre los puntos de 1

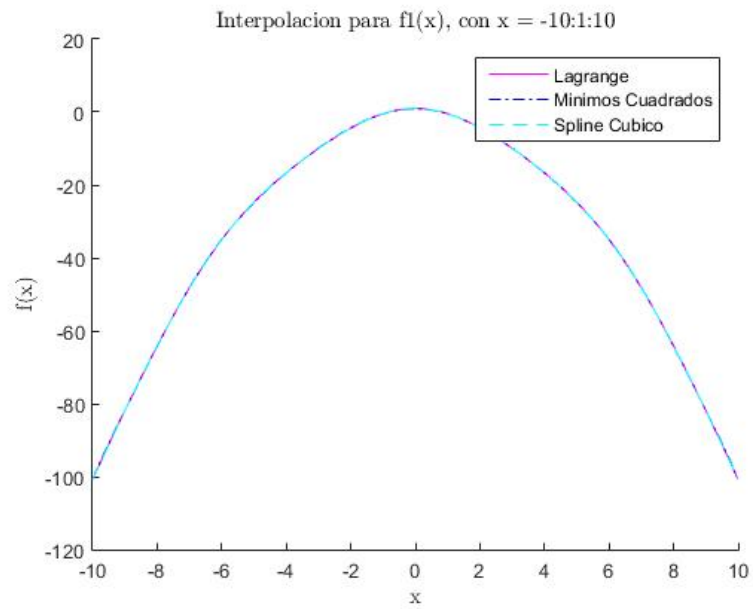


Figura 3-12: Interpolación para $f_1(x)$, con diferencia de 1 entre los puntos

Tabla 3.9: Error RMSE en comparación con valores reales de la función f_{1x}

Método	Error RMSE
Lagrange	1.654767982377897e-04
Mínimos Cuadrados	0.107281947329000
Spline Cúbico	0.011771783956323

3.2.1.4 Diferencia entre los puntos de 0.5

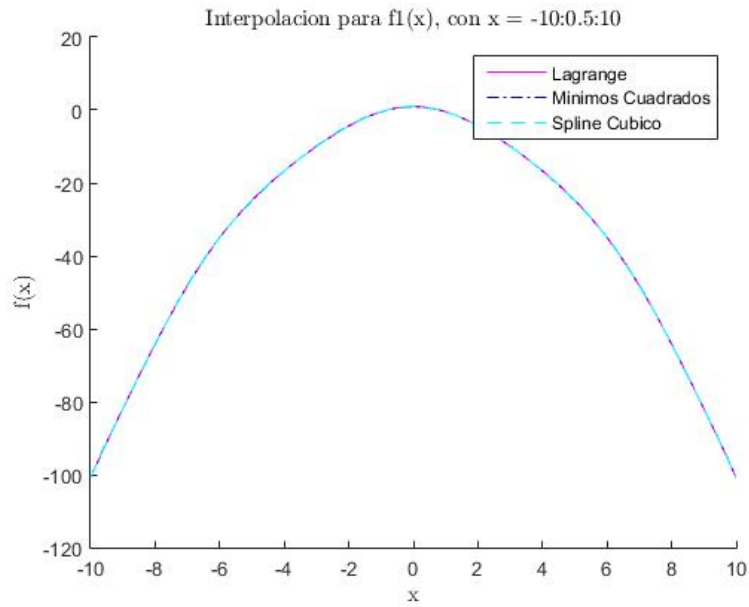


Figura 3-13: Interpolación para $f_1(x)$, con diferencia de 0.5 entre los puntos

Tabla 3.10: Error RMSE en comparación con valores reales de la función $f_1(x)$

Método	Error RMSE
Lagrange	1.070689047846483e-07
Mínimos Cuadrados	0.094762812417601
Spline Cúbico	0.002250071745204

3.2.2 $f_2(x)$

Recordar que $f_2(x) = x^3 + 2 * x^2 - 5$

3.2.2.5 Diferencia entre los puntos de 5

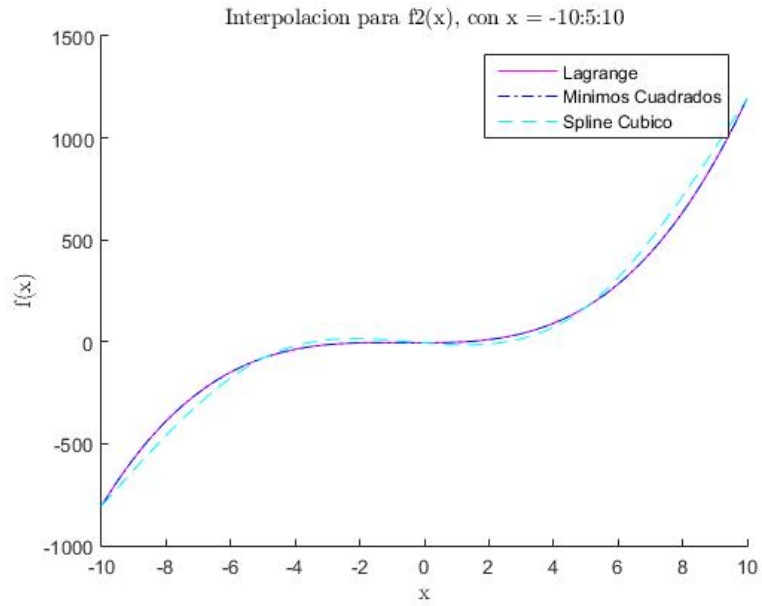


Figura 3-14: Interpolación para $f_2(x)$, con diferencia de 5 entre los puntos

Tabla 3.11: Error RMSE en comparación con valores reales de la función $f_{2,x}$

Método	Error RMSE
Lagrange	5.482145877462961e-14
Mínimos Cuadrados	8.835681654582793e-14
Spline Cúbico	39.080816519445406

3.2.2.6 Diferencia entre los puntos de 2

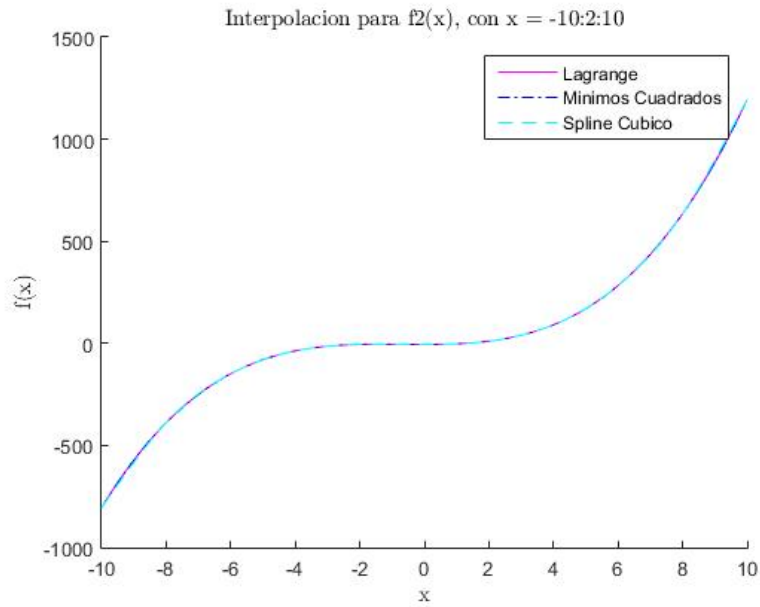


Figura 3-15: Interpolación para $f_2(x)$, con diferencia de 2 entre los puntos

Tabla 3.12: Error RMSE en comparación con valores reales de la función $f_2(x)$

Método	Error RMSE
Lagrange	9.837321126766994e-14
Mínimos Cuadrados	4.285649511124380e-14
Spline Cúbico	3.813830254035614

3.2.2.7 Diferencia entre los puntos de 1

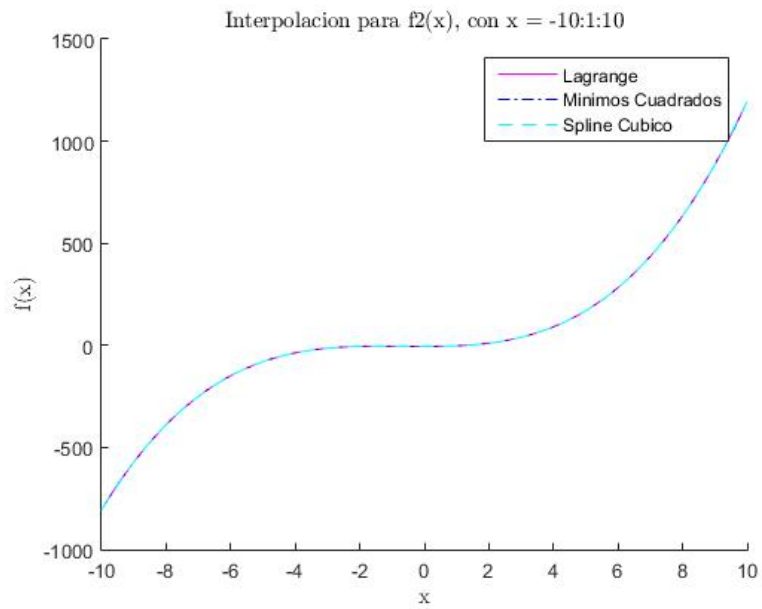


Figura 3-16: Interpolación para $f_2(x)$, con diferencia de 1 entre los puntos

Tabla 3.13: Error RMSE en comparación con valores reales de la función $f_{2,x}$

Método	Error RMSE
Lagrange	2.018848590001712e-12
Mínimos Cuadrados	9.194153009378197e-12
Spline Cúbico	0.674171195633541

3.2.2.8 Diferencia entre los puntos de 0.5

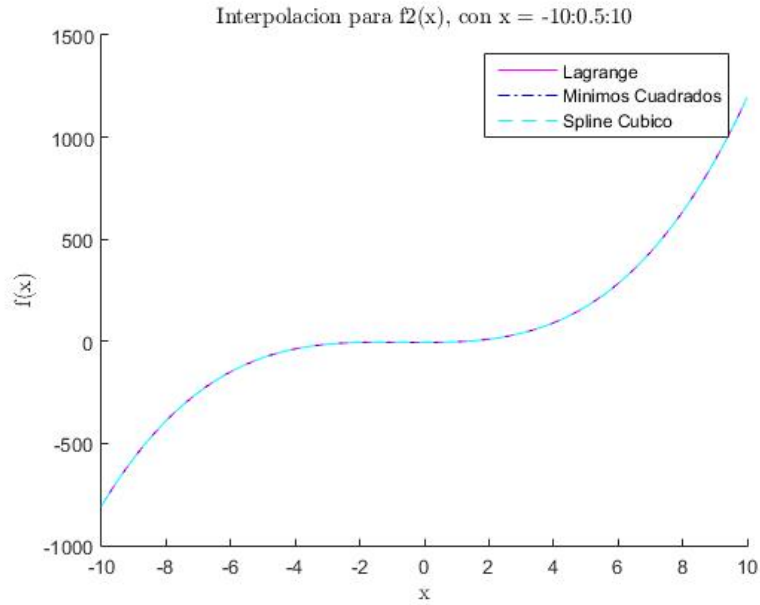


Figura 3-17: Interpolación para $f_2(x)$, con diferencia de 0.5 entre los puntos

Tabla 3.14: Error RMSE en comparación con valores reales de la función $f_2(x)$

Método	Error RMSE
Lagrange	3.133123364401713e-07
Mínimos Cuadrados	7.707456727628345e-12
Spline Cúbico	0.119167143685748

CAPÍTULO 4. ANÁLISIS DE LOS RESULTADOS

4.1 MÉTODOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES LINEALES

A través de la obtención de las raíces para cada uno de los métodos, para cada uno de los espectros estudiados y al observar los gráficos de cada uno de los espectros, se encontraron diferencias mínimas las cuales no se pueden observar a plena vista, es por esto que se tomo la decisión de no incluir todos estos, sino que más bien, incluir uno que represente en forma general a las raíces obtenidas. Esto se afirma al observar los errores obtenidos en la comparación de las raíces obtenidas mediante los métodos implementados, con el método 'linsolve' de MATLAB, apreciando así, que el mayor error se presenta para lo métodos iterativos, recordando que esto se debe a la tolerancia escogida.

Si se observa el número de operaciones que usa cada método se puede afirmar que aquel método más rápido es el QR, seguido por LU, SOR y Gauss-Seidel. Hay que tener en cuenta que el error obtenido para los método no iterativos, es menor que con los otros dos métodos, por lo que se podría decir que estos primeros son más efectivos.

Así mismo, y resaltando parte de la idea anterior, el método de SOR, para un w efectivo converge más rápido que el de Gauss-Seidel. Esto se afirma en la observación tanto del gráfico del error, tanto del gráfico del error con escala logarítmica. Hay que tener entonces, un especial cuidado a la hora de escoger el coeficiente de relajación w para el método SOR, ya que para otros valores, los resultados pueden variar, provocando una convergencia más lenta.

Cabe resaltar que para el espectro de 4225×4225 , a la hora de escoger un w adecuado, debido a la gran cantidad de operaciones y el tiempo de ejecución que este espectro conlleva, no se pudo realizar diferentes pruebas para definir un w , por lo tanto se tomó la decisión de tomar este con un valor de 1, obteniendo los mismos resultados que el método de Gauss-Seidel.

4.2 MÉTODOS DE INTERPOLACIÓN

En una primera observación de cada uno de los métodos implementados se puede observar que entre más puntos se le entregan a cada uno de estos, mejor es la aproximación del polinomio a los resultados reales.

Se puede observar así mismo, que dependiendo de los puntos que se le entregan a cada método, y dependiendo de la función estudiada, un método puede tener un resultado con un error menor que los otros. Pero al cabo de lo que se aprecia con todas las diferencias de puntos hechas, y con las diferentes funciones dadas, aquel que presenta un menor error, comparándose con los resultados reales, es el de Lagrange.

CAPÍTULO 5. CONCLUSIONES

Según los resultados obtenidos, y el análisis hecho, se puede concluir lo siguiente: Aunque se pudo implementar todos los métodos pedidos en MATLAB, y se obtuvieron los resultados que se presentan a lo largo del informe, existe cierta duda en la validez de los resultados obtenidos, un ejemplo claro de ello, se da en los resultados obtenidos en el error RMSE de la interpolación mediante Spline Cúbicos, ya que en un caso en específico este tiene un valor aproximado de 39, lo que genera cierta incertidumbre debido a que según lo aprendido y lo documentado, este método es uno de los más utilizados porque da lugar a resultados similares requiriendo solamente el uso de polinomios de bajo grado. Por lo tanto, se podría afirmar que de alguna manera u otra los objetivos impuestos para este laboratorio no fueron cumplidos a cabalidad.

Es importante así mismo la optimización de los códigos implementados, por lo que se debe buscar que el orden de los algoritmos sea el menor posible, garantizando que los resultados sean óptimos con un tiempo de ejecución bajo. Este fue uno de los puntos que mas trabajo costó en la realización de este laboratorio, ya que para el caso del espectro de 4225×4225 , se tuvo que mejorar el proceso de los algoritmos para poder obtener los resultados aquí presentados. Al final, para este caso, cada método se ejecutó con éxito en alrededor de un lapso entre treinta minutos a una hora.

Si se tuviera que ordenar los métodos de resolución para sistemas de ecuaciones lineales, en base a los resultados obtenidos, de mejor a peor, sería lo siguiente:

1. QR
2. LU
3. SOR (Con un w adecuado)
4. Gauss-Seidel

Tomando preferencia por los métodos directos, en contraste con los métodos iterativos. Y teniendo principal ventaja el método de QR, puesto que este se puede aplicar a matrices no cuadradas.

En el caso de los métodos de interpolación, en base a los resultados obtenidos, de mejor a peor, se puede considerar lo siguiente:

1. Lagrange
2. Mínimos Cuadrados
3. Spline Cúbico

Puede que según los puntos y la función a considerar, se intercambien de lugar el método de Mínimos Cuadrados y el Spline Cúbico

CAPÍTULO 6. BIBLIOGRAFÍA

Chapra, S. Canale, R. (2007). Método numéricos para ingeniería (5th ed.)

Cobos, F. U. d. S. (2002). Apuntes de cálculo numérico.