



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE CIENCIAS
CARRERA MATEMÁTICA

**SIMULACIÓN NUMÉRICA PARA EL PARÁMETRO ÓPTIMO DE
RELAJACIÓN DEL MÉTODO SOR APLICADO A UN ESQUEMA
DE DIFERENCIAS FINITAS ASOCIADO A LA ECUACIÓN DE
POISSON**

Trabajo de Integración Curricular

Tipo: Proyecto de Investigación

Presentado para optar al grado académico de:

MATEMÁTICO

AUTOR:

ALEX YASSER YAGUAL GONZÁLEZ

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE CIENCIAS
CARRERA MATEMÁTICA

**SIMULACIÓN NUMÉRICA PARA EL PARÁMETRO ÓPTIMO DE
RELAJACIÓN DEL MÉTODO SOR APLICADO A UN ESQUEMA
DE DIFERENCIAS FINITAS ASOCIADO A LA ECUACIÓN DE
POISSON**

Trabajo de Integración Curricular

Tipo: Proyecto de Investigación

Presentado para optar al grado académico de:

MATEMÁTICO

AUTOR: ALEX YASSER YAGUAL GONZÁLEZ

DIRECTOR: ING. ALEX EDUARDO POZO VALDIVIEZO, MSc.

Riobamba – Ecuador

2023

©2023, Alex Yasser Yagual González

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, **ALEX YASSER YAGUAL GONZÁLEZ**, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 10 de noviembre de 2023



Alex Yasser Yagual González

C.I. 2400358913

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE CIENCIAS
CARRERA MATEMÁTICA

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; tipo: Proyecto de Investigación, **SIMULACIÓN NUMÉRICA PARA EL PARÁMETRO ÓPTIMO DE RELAJACIÓN DEL MÉTODO SOR APLICADO A UN ESQUEMA DE DIFERENCIAS FINITAS ASOCIADO A LA ECUACIÓN DE POISSON**, realizado por el señor: **ALEX YASSER YAGUAL GONZÁLEZ**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

	FIRMA	FECHA
Ing. Maria José Mendoza Salazar, Mgs. PRESIDENTE DEL TRIBUNAL	 _____	2023-11-10 _____
Ing. Alex Eduardo Pozo Valdiviezo, MSc. DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR	 _____	2023-11-10 _____
Dra. Martha Ximena Dávalos Villegas ASESORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR	 _____	2023-11-10 _____

DEDICATORIA

Este trabajo de integración curricular se lo dedico en primer lugar a Dios por ser eje fundamental en mi desarrollo y crecimiento de vida, por darme la fortaleza y sabiduría para concluir mi etapa universitaria. En segundo lugar a mis amados padres quienes con su amor, esfuerzo, consejos y paciencia han sabido guiar cada paso en la culminación exitosa de mi formación académica universitaria. Y para finalizar a mis familiares y amigos que me acompañaron de una manera u otra en transcurso de este período.

Alex

AGRADECIMIENTO

A mi padre Aníbal, por ser mi fuente de inspiración constante, motivándome a avanzar y transmitiéndome su aprecio por las matemáticas, a mi querida madre Marjorie, por su apoyo incondicional a lo largo de este arduo proceso. Expreso mi eterna gratitud a mi Director de Tesis, Eduardo Pozo, cuya dedicación, sabiduría, paciencia y motivación constante fueron fundamentales para esta investigación. Agradezco a mis amigos Anabel y Jeremy por los buenos momentos compartidos en nuestra vida universitaria. Vuestra amistad ha sido un pilar fundamental. A todos ustedes, mi más sincero agradecimiento por su apoyo inquebrantable y contribución significativa para alcanzar este logro académico. Sin su aliento, este camino habría sido mucho más difícil. Gracias por ser parte de mi vida y compartir esta travesía conmigo.

Alex

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS	ix
ÍNDICE DE ILUSTRACIONES	x
ÍNDICE DE ANEXOS	xi
RESUMEN	xii
ABSTRACT	xiii
INTRODUCCIÓN	1

CAPÍTULO I

1. PROBLEMA DE INVESTIGACIÓN	3
1.1. Planteamiento del problema	3
1.2. Objetivos	3
1.2.1. <i>Objetivo General</i>	3
1.2.2. <i>Objetivos específicos</i>	4
1.3. Justificación	4

CAPÍTULO II

2. MARCO TEORICO	6
2.1. Bases teóricas	6
2.1.1. <i>Álgebra Matricial</i>	7
2.1.2. <i>Sistemas de ecuaciones lineales</i>	10
2.1.3. <i>Diferentes tipos de sistemas lineales</i>	12
2.1.4. <i>Condiciones para la existencia de solución única</i>	12
2.2. Solución numérica del sistema de ecuaciones lineales $A\vec{x} = \vec{b}$ por medio de métodos iterativos	13
2.2.1. <i>Construcción de métodos iterativos</i>	15
2.2.2. <i>Método de Jacobi</i>	16
2.2.3. <i>Método de Gauss-Seidel</i>	17
2.2.4. <i>Método de Sobrerelajacion Sucesiva</i>	20

CAPÍTULO III

3.	MARCO METODOLÓGICO	26
3.1.	Descripción de enfoque, alcance, diseño, tipo, métodos, técnicas e instrumentos de investigación empleadas	26

CAPÍTULO IV

4.	MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	27
4.1.	Parámetro óptimo de relajación del Método SOR.	27
4.2.	Discretización de la ecuación de Poisson	30
4.3.	Simulación numérica de la ecuación de Poisson.	37

CAPÍTULO V

5.	CONCLUSIONES Y RECOMENDACIONES	44
5.1.	CONCLUSIONES	44
5.2.	RECOMENDACIONES	45

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 4-1: Errores para cada método iterativo	40
Tabla 4-2: Análisis del error en métodos iterativos	40
Tabla 4-3: Influencia del ω en el error SOR	42
Tabla 4-4: Parámetro óptimo del método SOR	43

ÍNDICE DE ILUSTRACIONES

Ilustración 4-1: Análisis de métodos iterativos para la ecuación de Poisson	39
Ilustración 4-2: Error en métodos iterativos para la ecuación de Poisson.	39
Ilustración 4-3: Análisis del error relativo entre métodos iterativos	40
Ilustración 4-4: Análisis de la elección del parámetro ω en la solución SOR	41
Ilustración 4-5: Error del método iterativo SOR para diferentes valores de ω	41
Ilustración 4-6: Análisis del ω_{opt} para el método iterativo SOR	42
Ilustración 4-7: Error del método iterativo SOR para el ω_{opt}	43

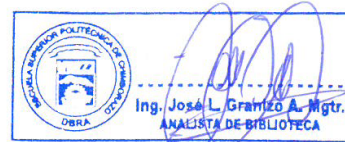
ÍNDICE DE ANEXOS

**ANEXO A: MÉTODOS ITERATIVOS PARA LA RESOLUCIÓN DE LA ECUACIÓN DE
POISSON**

RESUMEN

El curso de Análisis Numérico en la ESPOCH se centra en métodos clásicos para resolver ecuaciones lineales, siendo menos eficientes para sistemas grandes y complejos. La falta de información detallada sobre métodos iterativos, como el SOR, limita su aplicabilidad en investigaciones y tecnologías, por lo tanto, el objetivo de este trabajo consistió en llevar a cabo una investigación documental de carácter teórico-práctico sobre el parámetro óptimo de relajación del método SOR aplicado a un esquema de diferencias finitas asociado a la ecuación de Poisson. La metodología se basó en un enfoque cualitativo y nivel descriptivo, centrado en la revisión bibliográfica de fuentes secundarias. Particularmente, se examinaron documentos concernientes a la optimización del parámetro en el método SOR, enfocados en sistemas de ecuaciones derivados de la discretización en diferencias finitas de la ecuación de Poisson. Los análisis y simulaciones realizados evidenciaron que el valor óptimo del parámetro de relajación del método SOR aumenta a medida que se incrementa el número de subdivisiones, cabe mencionar que este valor debe garantizar la convergencia del método hacia la solución. Asimismo, se destaca también la necesidad de aumentar la tolerancia y el número de iteraciones en sistemas más complejos para lograr resultados precisos. En conclusión, tras realizar un análisis comparativo entre los métodos iterativos de Jacobi, Gauss-Seidel y SOR, se establece que el método SOR sobresale por su velocidad de convergencia y precisión en la simulación numérica del parámetro óptimo de relajación. La elección adecuada de este parámetro resulta esencial para obtener soluciones precisas en sistemas de ecuaciones lineales extensos, realizando así la eficiencia y exactitud del método.

Palabras clave: <MÉTODOS ITERATIVOS>, <PARÁMETRO ÓPTIMO DE RELAJACIÓN>, <ECUACIÓN DE POISSON>, <DIFERENCIAS FINITAS>.



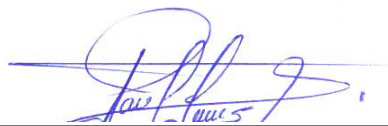
28-11-2023

1998-DBRA-UPT-2023

ABSTRACT

The Numerical Analysis course carried out at ESPOCH focuses on classical methods for solving linear equations, but at the same time they are not efficient for solving large and complex systems. The lack of detailed information on iterative methods, such as SOR, limits its applicability in research and technologies; therefore, the aim of this work was to carry out a theoretical-practical documentary research on the optimal relaxation parameter of the SOR method applied to a finite difference scheme which was associated to Poisson's equation. The methodology was based on a qualitative approach and descriptive level, focused on the bibliographic review of secondary sources. Particularly, documents concerning the optimization of the parameter in the SOR method were examined. This parameter was based on systems of equations derived from the finite difference discretization of the Poisson equation. The analyses and simulations carried out evidenced that the optimal value of the relaxation parameter of the SOR method increases as the number of subdivisions increases, thus it is important to mention that this value must guarantee the convergence of the method towards the solution of the system. Likewise, the need to increase the tolerance and the number of iterations in more complex systems to achieve accurate results is also highlighted. In conclusion, after performing a comparative analysis between the Jacobi, Gauss-Seidel and SOR iterative methods, it is established that the SOR method stands out due to its convergence speed and accuracy in the numerical simulation of the optimal relaxation parameter. The appropriate choice of this parameter is essential in the obtention of accurate solutions in large linear equation systems, improving the efficiency and accuracy of the method.

Keywords: : <ITERATIVE METHODS>, <OPTIMAL RELAXATION PARAMETER>, <POISSON EQUATION>, <FINITE DIFFERENCES>.



Lic. Paul Rolando Armas Pesantez Mgs.

060328987-7

INTRODUCCIÓN

Los métodos iterativos permiten hallar la solución aproximada de un sistema de ecuaciones lineales, construyendo una sucesión de aproximaciones, a partir de una aproximación inicial, en particular nos enfocaremos en el método de sobrerelajación sucesiva (SOR por sus siglas en inglés).

El método SOR fue desarrollado simultáneamente por David M. Young Jr. y Stanley P. Frankel en la década de 1950, es uno de los algoritmos iterativos más populares, el cual ha sido ampliamente utilizado en diversas áreas de la ciencia debido a su eficiencia y capacidad para resolver problemas que involucran ecuaciones lineales de manera rápida y precisa. Young (1954) mostró que para una elección adecuada de este parámetro se obtiene una aceleración significativa de la convergencia de iteraciones (p. 92); mientras que, Frankel (1950) demostró que este método es particularmente efectivo en la resolución de problemas que involucran matrices diagonales dominantes, simétricas y definidas positivas (p. 65).

En este contexto, se aborda la resolución de sistemas de ecuaciones lineales de gran tamaño, a menudo provenientes de la discretización de ecuaciones diferenciales por diferencias finitas, particularmente consideraremos el sistema derivado de la discretización de la ecuación de Poisson y a su vez se elegirá un parámetro óptimo de relajación para garantizar una solución precisa y una rápida convergencia del método.

El capítulo 1, está organizado por la problemática que dio origen a esta investigación, delineando los objetivos que guiarán su desarrollo. Además, se expone la justificación de este estudio, enfatizando su relevancia en el ámbito del Análisis Numérico y su aplicación práctica. El capítulo 2, está constituido por el fundamento matemático esencial para abordar el método SOR, así como de la optimización del parámetro ω en dicho contexto. El capítulo 3, está compuesto por la metodología seguida en el desarrollo de este trabajo.

El capítulo 4, se muestra los resultados obtenidos a partir de las simulaciones numéricas realizadas, centrándose en la convergencia del método SOR para diferentes valores del parámetro de relajación ω , con el objetivo de determinar el valor que minimiza el error y maximiza la eficiencia del método. Se compararán los resultados obtenidos con otros métodos iterativos tal como es de: Jacobi, Gauss-Seidel utilizados para resolver la ecuación de Poisson y se discutirán cual es el más

idóneo para obtener soluciones precisas de este sistema de ecuaciones.

Finalmente, en el capítulo 5 se presentan las conclusiones obtenidas a partir del trabajo, de igual forma, se ofrecen recomendaciones para investigaciones futuras en este ámbito de estudio.

CAPÍTULO I

1. PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema

El curso de Análisis Numérico de la carrera de Matemáticas en la ESPOCH se enfoca principalmente en los métodos clásicos para la resolución de sistemas de ecuaciones lineales, como el método de Gauss y Factorización LU. Aunque estos métodos son altamente efectivos en la resolución de pequeños sistemas de ecuaciones lineales, no son óptimos para la solución de sistemas grandes y complejos. En este sentido, los métodos iterativos surgen como una alternativa valiosa y eficiente para la resolución de sistemas de ecuaciones ya que requieren menos memoria que los métodos directos, son más rápidos y pueden manejar estructuras especiales (como matrices tridiagonales) de una manera simple.

No obstante, la carencia de información detallada y accesible acerca de la resolución de sistemas de ecuaciones lineales a través de métodos iterativos en el entorno académico como su implementación y optimización en aplicaciones prácticas. La falta de claridad en aspectos clave, como la elección óptima del factor de relajación, la determinación de criterios de convergencia y la adaptación del método a sistemas de ecuaciones más complejos, por ejemplo, sistemas de ecuaciones provenientes de la discretización en diferencias finitas de la ecuación de Poisson, limita la eficacia y el alcance del método SOR. Por lo tanto, abordar este problema de falta de documentación se vuelve esencial para aprovechar todo el potencial que el método SOR podría ofrecer en diversas áreas de investigación y aplicaciones tecnológicas.

1.2. Objetivos

1.2.1. *Objetivo General*

Realizar una investigación documental de carácter teórico-práctico sobre el parámetro óptimo de relajación del método SOR aplicado a un esquema de diferencias finitas asociado a la ecuación de Poisson, el cual será implementado en Python para diseñar un documento que sirva de guía para entender la aplicación de este método.

1.2.2. *Objetivos específicos*

- Realizar una búsqueda sistemática y selectiva de referencias pertinentes.
- Profundizar en el método SOR, abarcando su comprensión de fundamentos teóricos, identificación de criterios de convergencia y determinación del óptimo parámetro de relajación.
- Emplear el lenguaje de programación Python para implementar el método SOR en la resolución de los esquemas en diferencias finitas correspondientes a la ecuación de Poisson.
- Llevar a cabo un análisis exhaustivo de los resultados obtenidos.

1.3. **Justificación**

La resolución discreta de la ecuación de Poisson desempeña un papel fundamental en numerosos campos, como la modelización de fenómenos electromagnéticos, la dinámica de fluidos y la simulación de sistemas físicos complejos. En este contexto, el método SOR (Successive Over-Relaxation) ha demostrado ser una herramienta potencialmente valiosa para abordar esta ecuación en su forma discreta. Mas aún, en la Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Carrera de Matemática, los estudiantes carecen de documentos en donde se explique como a través del método iterativo SOR se puede resolver sistemas de ecuaciones de gran tamaño y a su vez el análisis para optimizar el parámetro de relajación ω que permita acelerar la convergencia del método.

Aunque el método SOR ha demostrado ser una herramienta valiosa en diversas disciplinas, desde la ingeniería hasta la física teórica, la falta de recursos claros y estructurados que aborden sus fundamentos y aplicaciones prácticas dificulta su estudio exhaustivo. La escasez de manuscritos y literatura especializada sobre el método SOR limita la capacidad de los y las estudiantes para manipular este método numérico, tratando de comprender muy atentamente los aspectos cruciales, como la selección óptima del factor de relajación, la evaluación de la convergencia y la adaptación a sistemas de ecuaciones más complejos, crea barreras para su incorporación funcional. En este contexto, resulta imperativo abordar el problema de la falta de documentación del método SOR para permitir un estudio más completo y fructífero de sus aplicaciones, maximizando así su utilidad en diversas áreas de investigación y práctica tecnológica.

Por lo tanto, es crucial abordar el problema de la falta de documentación del método SOR en la

resolución discreta de la ecuación de Poisson. Al proporcionar recursos claros, ejemplos prácticos y directrices detalladas, se facilitaría tanto la comprensión teórica como la implementación efectiva de este método en un contexto tan relevante. Esto, a su vez, promovería un avance significativo en la capacidad de resolver de manera eficiente y precisa la ecuación de Poisson en su forma discreta, con aplicaciones presentes en la literatura, maximizando así su utilidad en diversas áreas de investigación y práctica tecnológica.

CAPÍTULO II

2. MARCO TEÓRICO

El método de sobrerrelajación sucesiva es uno de los métodos iterativos más eficientes y ampliamente utilizado para resolver sistema de ecuaciones lineales de la forma $A\vec{x} = \vec{b}$, donde la matriz $A \in M_{n \times n}(\mathbb{R})$ es invertible y $\vec{x}, \vec{b} \in \mathbb{R}^n$, que se expresa a través de la siguiente relación de recurrencia:

$$x^{(k+1)} = (D + \omega L)^{-1}[(1 - \omega)D - \omega U]x^{(k)} + \omega(D + \omega L)^{-1}b, \quad k \in \mathbb{N}, \quad (2.1)$$

donde D es la diagonal de A , L es una matriz triangular inferior, U es una matriz triangular superior, y $\omega \in \mathbb{R}^+$ representa el parámetro de sobrerrelajación que se utiliza para acelerar la convergencia del método.

La elección adecuada del parámetro de relajación mejora significativamente la eficiencia y precisión del método. En particular, para una matriz A de un sistema general, el método SOR converge cuando $0 < \omega < 2$, esto implica que existe un valor óptimo de ω que maximiza la velocidad y la estabilidad de la convergencia del método el cual viene dado por:

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \beta^2}}, \quad (2.2)$$

donde β es el radio espectral de la matriz iteración del método SOR, y las satisface las condiciones de relaciones de recurrencia dada en (2.1).

La información que se presenta a continuación se ha extraído de las siguientes fuentes académicas: (Benalcázar, 2007, Grossman et al., 2019, Faires y Burden, 2012).

2.1. Bases teóricas

Estableceremos una notación matemática consistente a lo largo de este documento, utilizaremos símbolos y convenciones estándar ampliamente reconocidos en el campo de las matemáticas. Esta notación nos permitirá expresar de manera clara y concisa conceptos matemáticos a lo largo del documento. Introducimos la notación \vec{x} para los elementos de \mathbb{R}^n , al conjunto de matrices cuadradas de orden n , representaremos como $M_{n \times n}(\mathbb{R})$ y $A = (a_{ij})_{n \times n}$ una matriz cuadrada,

mientras que u' y u'' denotan las primeras y segundas derivadas de la función u respecto a la variable x , respectivamente, a su vez, el radio espectral de una matriz A viene representado por $\rho(A)$. Además, empleamos $\|\cdot\|_\infty$ para representar la norma infinito para matrices, definida como

$$\|A\|_\infty = \max_{i=1,2,\dots,n} \sum_{j=1}^n |a_{ij}|$$

2.1.1. Álgebra Matricial

En el ámbito de las matemáticas, en particular, el Análisis Numérico, el Álgebra Matricial se eleva como una herramienta fundamental para analizar sistemas de ecuaciones lineales y resolver una amplia gama de problemas provenientes de diversos métodos numéricos. En esencia, el Álgebra Matricial se centra en el estudio de matrices y sus operaciones, donde una matriz es una disposición rectangular de números o elementos organizados en filas y columnas. Estos elementos pueden representar desde coeficientes en sistemas de ecuaciones hasta características en conjuntos de datos multidimensionales.

El Álgebra Matricial ofrece un fundamento sólido para realizar operaciones como la suma y multiplicación de matrices, el cálculo de determinantes, la inversión de matrices y la resolución de sistemas de ecuaciones lineales. Además de su valor teórico, el Álgebra Matricial encuentra una amplia aplicación en campos como la física, la ingeniería, la economía, la estadística y la informática, ya que proporciona una manera efectiva de abordar problemas complejos y analizar relaciones entre variables.

En las subsiguientes partes, exploraremos los conceptos básicos del Álgebra Matricial, desde la definición y notación de matrices hasta las operaciones clave que permiten comprender cómo estas estructuras numéricas pueden ser manipuladas y utilizadas para abordar el método SOR.

Notemos que, existen diversos tipos de matrices detalladas a continuación.

Definición 2.1. Sea $A \in M_{n \times n}(\mathbb{R})$. Decimos que A es una matriz diagonal si

$$a_{ij} = 0, \quad \forall i \neq j.$$

Definición 2.2. Sea $A \in M_{n \times n}(\mathbb{R})$. Decimos que A es una matriz triangular inferior si

$$a_{ij} = 0, \quad \forall i < j.$$

Donde A se visualiza como

$$A = \begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ a_{n,1} & \cdots & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

Definición 2.3. Sea $A \in M_{n \times n}(\mathbb{R})$. Decimos que A es una matriz triangular superior si

$$a_{ij} = 0, \quad \forall i > j.$$

Donde la matriz A se visualiza como

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1,n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2,n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & a_{n,n} \end{bmatrix}.$$

Definición 2.4. Sea $A \in M_{n \times n}(\mathbb{R})$. Decimos que A es una matriz tridiagonal si

$$a_{ij} = 0, \quad |i - j| > 1.$$

Así, la matriz tridiagonal viene dado por:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

Definición 2.5. Sea $A \in M_{n \times n}(\mathbb{R})$. Decimos que A es diagonalmente dominante si y solo si

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|, \quad \forall i = 1, \dots, n.$$

Definición 2.6. Sea $A \in M_{n \times n}(\mathbb{R})$. Se dice que A es una matriz estrictamente diagonalmente

dominante si

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|, \quad \forall i = 1, \dots, n.$$

Definición 2.7. Sea $A \in M_{n \times n}(\mathbb{R})$. Se dice que A es simétrica si

$$A = A^\top$$

Definición 2.8. Sea $A \in M_{n \times n}(\mathbb{R})$. Se dice que A es definida positiva si para todo $\vec{x} \neq \vec{0}$ se tiene que

$$\vec{x}^\top A \vec{x} > 0.$$

Es importante destacar que el criterio de Sylvester resulta ser una herramienta más práctica para analizar cuando una matriz es (semi)definida positiva; puesto que usa los menores principales de la matriz.

Definición 2.9. Sea $A \in M_{n \times n}(\mathbb{R})$. Se dice que A es diagonalizable, si existe una matriz invertible $Q \in M_{n \times n}(\mathbb{R})$ tal que

$$D = Q^{-1}AQ. \quad (2.3)$$

Definición 2.10. Sea $A \in M_{n \times n}(\mathbb{R})$. Se dice que A es ortogonal si A es invertible y $A^{-1} = A^\top$.

Definición 2.11. Sea $A \in M_{n \times n}(\mathbb{R})$. Se dice que A es ortogonalmente diagonalizable si existe una matriz ortogonal $Q \in M_{n \times n}(\mathbb{R})$ tal que

$$Q^\top A Q = D, \quad (2.4)$$

donde $D = \text{diag}(\lambda_i)$ y $\lambda_i, i = 1, \dots, n$ es un valor propio de A .

Con el propósito de ejemplificar las definiciones que hemos analizado previamente, notemos el siguiente ejemplo.

Ejemplo. Consideremos $A \in M_{n \times n}(\mathbb{R})$ dada por

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

Observemos que, en efecto, A es simétrica y diagonalmente dominante, mostremos que es definida positiva. Para ello, consideramos un vector $\vec{x} \in \mathbb{R}^4$ no nulo.

$$\begin{aligned} \vec{x}^T A \vec{x} &= \begin{bmatrix} \vec{x}_1 & \vec{x}_2 & \vec{x}_3 & \vec{x}_4 \end{bmatrix} \begin{bmatrix} 2\vec{x}_1 - \vec{x}_2 \\ -\vec{x}_1 + 2\vec{x}_2 - \vec{x}_3 \\ -\vec{x}_2 + 2\vec{x}_3 - \vec{x}_4 \\ -\vec{x}_3 + 2\vec{x}_4 \end{bmatrix} \\ &= (2\vec{x}_1 - \vec{x}_2)\vec{x}_1 + (-\vec{x}_1 + 2\vec{x}_2 - \vec{x}_3)\vec{x}_2 + (-\vec{x}_2 + 2\vec{x}_3 - \vec{x}_4)\vec{x}_3 + (-\vec{x}_3 + 2\vec{x}_4)\vec{x}_4 \\ &= 2\vec{x}_1^2 - \vec{x}_1\vec{x}_2 - \vec{x}_1\vec{x}_2 + 2\vec{x}_2^2 - \vec{x}_2\vec{x}_3 - \vec{x}_2\vec{x}_3 + 2\vec{x}_3^2 - \vec{x}_3\vec{x}_4 - \vec{x}_3\vec{x}_4 + 2\vec{x}_4^2 \\ &= 2\vec{x}_1^2 - 2\vec{x}_1\vec{x}_2 + 2\vec{x}_2^2 - 2\vec{x}_2\vec{x}_3 + 2\vec{x}_3^2 - 2\vec{x}_3\vec{x}_4 + 2\vec{x}_4^2 \\ &= (\vec{x}_1 - \vec{x}_2)^2 + (\vec{x}_2 - \vec{x}_3)^2 + (\vec{x}_3 - \vec{x}_4)^2 + \vec{x}_1^2 + \vec{x}_4^2 > 0 \end{aligned}$$

Por lo tanto, A es definida positiva.

Comprobemos si la matriz A es ortogonal:

$$\begin{aligned} AA^T &= \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & -4 & 1 & 0 \\ -4 & 6 & -4 & 1 \\ 1 & -4 & 6 & -4 \\ 0 & 1 & -4 & 5 \end{bmatrix} \\ A^T A &= \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & -4 & 1 & 0 \\ -4 & 6 & -4 & 1 \\ 1 & -4 & 6 & -4 \\ 0 & 1 & -4 & 5 \end{bmatrix} \end{aligned}$$

Podemos notar que $AA^T = A^T A \neq I$, entonces se sigue que A no es ortogonal.

Una vez que hemos resumido aspectos relevantes del Álgebra Matricial, pasamos a su relación con el estudio de sistemas de ecuaciones lineales.

2.1.2. Sistemas de ecuaciones lineales

En esta sección, exploraremos la teoría, métodos de solución y aplicaciones de los sistemas de ecuaciones lineales, dilucidando su importancia y adaptabilidad en diversas aplicaciones.

Definición 2.12. Denominamos ecuación lineal en las variables (o incógnitas) x_1, \dots, x_n a toda

relación de la forma

$$a_1x_1 + \cdots + a_nx_n = b, \quad (2.5)$$

donde a_1, \dots, a_n y b son números reales dados.

Definición 2.13. Un sistema de n ecuaciones lineales con m incógnitas es una lista de n ecuaciones lineales.

Usualmente escribimos tales sistemas en n líneas colocadas unas bajo las otras. La forma general de un sistema lineal de n ecuaciones con m incógnitas es la siguiente:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m = b_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m = b_n \end{cases} \quad (2.6)$$

Los números a_{ij} , $i = 1, \dots, n$, $j = 1, \dots, m$, son los coeficientes del sistema. Estos son datos. Los números b_i , $i = 1, \dots, n$, constituyen el segundo miembro del sistema y también son datos.

Es recomendable observar el sistema (2.6) bajo la forma matricial $A\vec{x} = \vec{b}$ con

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

donde $A \in M_{n \times m}(\mathbb{R})$, $\vec{x} \in \mathbb{R}^m$ y $\vec{b} \in \mathbb{R}^n$.

Definición 2.14. Una solución del sistema lineal es una lista de m números reales (s_1, s_2, \dots, s_p) tales que si sustituimos s_1 por x_1 , s_2 por x_2 y así sucesivamente, en el sistema lineal, obtenemos una igualdad. El conjunto de las soluciones del sistema es el conjunto de todas estas listas.

Como principio general, nos centramos en determinar el conjunto de soluciones de un sistema lineal. Esto se denomina resolver el sistema lineal. Esto conduce a la siguiente definición.

Definición 2.15. Se dice que dos sistemas lineales son equivalentes si tienen el mismo conjunto de soluciones.

A partir de aquí, resolver un sistema lineal determinado consistirá en transformarlo en un sistema equivalente cuya resolución será más sencilla que la del sistema inicial, como es el caso de la eliminación gaussiana.

2.1.3. Diferentes tipos de sistemas lineales

Enunciaremos un resultado teórico relevante para sistemas lineales.

Teorema 2.1. *Un sistema de ecuaciones lineales no tiene solución, tiene una solución única o tiene infinitas soluciones.*

Un sistema lineal que no tiene solución se dice que es **incompatible**. La demostración de este teorema se encuentra en (Strang, 1988, pp. 95-96).

2.1.4. Condiciones para la existencia de solución única

Consideramos el caso donde el número de ecuaciones es igual al número de incógnitas:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\vec{b}}.$$

Entonces, $A \in M_{n \times n}(\mathbb{R})$ es una matriz cuadrada y $\vec{b} \in \mathbb{R}^n$. A continuación podemos utilizar las matrices para encontrar la solución del sistema lineal.

Teorema 2.2. *Si la matriz A es invertible, entonces la solución del sistema $A\vec{x} = \vec{b}$ es única y es:*

$$\vec{x} = A^{-1}\vec{b}$$

Demostración. Supongamos que la matriz A es invertible, por ende existe A^{-1} matriz inversa de A tal que $A^{-1}A = I$. Seguidamente,

$$A^{-1}(A\vec{x}) = A^{-1}\vec{b},$$

$$(A^{-1}A)\vec{x} = A^{-1}\vec{b},$$

$$I\vec{x} = A^{-1}\vec{b},$$

$$\vec{x} = A^{-1}\vec{b}.$$

A su vez, verifiquemos que \vec{x} es la solución del sistema de ecuaciones lineales $A\vec{x} = \vec{b}$. Efectivamente,

$$A\vec{x} = A(A^{-1}\vec{b}) = (AA^{-1})\vec{b} = I\vec{b} = \vec{b}.$$

■

Cabe notar que el sistema de ecuaciones lineales $A\vec{x} = \vec{b}$ tiene solución única si y solamente si el determinante de la matriz A es no nulo. Mientras que si la matriz A no es invertible, por ende o no hay solución, o hay infinitas soluciones.

El teorema que se presenta a continuación proporciona los criterios suficientes para asegurar la invertibilidad de una matriz. Esta propiedad, a su vez, simplifica la tarea de determinar si un sistema de ecuaciones lineales específico tiene solución o no.

Teorema 2.3. *Sea $A \in M_{n \times n}(\mathbb{R})$. Se tienen las siguientes afirmaciones:*

1. *Si A es estrictamente diagonalmente dominante, entonces A es invertible.*
2. *Si A es estrictamente diagonalmente dominante y simétrica con $a_{ii} > 0 \ i = 1, \dots, n$; entonces A es definida positiva.*

Demostración. La demostración de este teorema se encuentra en (Benalcázar, 2007, p. 333). ■

2.2. Solución numérica del sistema de ecuaciones lineales $A\vec{x} = \vec{b}$ por medio de métodos iterativos

Los métodos iterativos para sistemas de ecuaciones lineales son herramientas fundamentales en el campo del Análisis Numérico y la resolución de problemas en diversas disciplinas. Estos métodos ofrecen una opción eficiente y versátil a diferencia de los métodos directos, como la eliminación gaussiana o la factorización LU, al abordar sistemas de ecuaciones donde la matriz de coeficientes es grande, dispersa o mal condicionada. En cambio con los métodos directos, que obtienen la solución exacta en un número finito de pasos, los métodos iterativos mejoran gradualmente una aproximación inicial a través de un proceso recurrente. Esta naturaleza iterativa los convierte en herramientas valiosas para abordar problemas complejos y aprovechar al máximo los recursos computacionales disponibles. En este contexto, es esencial comprender el funcionamiento de estos métodos, sus ventajas y limitaciones, así como su aplicación en situaciones del mundo real que van desde la simulación de circuitos eléctricos hasta la modelización de fenómenos físicos y económicos (Quarteroni et al., 2007, p. 151).

Un método iterativo consiste en construir una sucesión de vectores $(\vec{x}^{(k)})_{k \in \mathbb{N}}$ de \mathbb{R}^n que converge a la solución exacta $\vec{x} \in \mathbb{R}^n$, es decir,

$$\lim_{k \rightarrow \infty} \vec{x}^{(k)} = \vec{x},$$

para cualquier vector inicial dado $\vec{x}^{(0)} \in \mathbb{R}^n$. Podemos escribir el sistema de ecuaciones en una forma iterativa como sigue:

$$\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{g}, \quad k \in \mathbb{N} \quad (2.7)$$

donde B es una matriz apropiada, dependiente de A , y \vec{g} es un vector adecuado, dependiente de A y \vec{b} , que deben cumplir la relación

$$\vec{x} = B\vec{x} + \vec{g}. \quad (2.8)$$

Esta última expresión corresponde a una forma equivalente al sistema original $A\vec{x} = \vec{b}$. En términos generales, los métodos iterativos reformulan este sistema, aprovechando la relación $\vec{x} = A^{-1}\vec{b}$. Por consiguiente, podemos expresar el vector \vec{g} de la siguiente manera:

$$\vec{g} = (I - B)A^{-1}\vec{b}.$$

Se define $\vec{e}^{(k)} := \vec{x} - \vec{x}^{(k)}$ como el error en la etapa k -ésima. Al restar (2.7) de (2.8), obtenemos:

$$\vec{x}^{(k+1)} - \vec{x} = B\vec{x}^{(k)} - B\vec{x} = B(\vec{x}^{(k)} - \vec{x}) = B\vec{e}^{(k)}$$

donde a B se la conoce como **matriz de iteración** asociada a (2.7), es importante mencionar que en \mathbb{R}^n , las normas $\|\cdot\|_p$ para $1 \leq p \leq \infty$ son equivalentes, lo que se expresa como:

$$\|x\|_\infty \leq \|x\|_p \leq \|x\|_1 \leq n\|x\|_\infty.$$

Esto asegura la coherencia de los resultados en el método iterativo, independientemente de la norma utilizada. A su vez, si B es una matriz simétrica y definida positiva, se arriba a, $\|B\vec{\omega}\| \leq \lambda_{\text{máx}}\|\vec{\omega}\|, \forall \vec{\omega} \in \mathbb{R}^n$ con $\lambda_{\text{máx}}$ el mayor valor propio de B , entonces tenemos

$$\|\vec{e}^{(k+1)}\| \leq \|B\vec{e}^{(k)}\| \leq \rho(B)\|\vec{e}^{(k)}\|, \quad \forall k \in \mathbb{N}$$

donde hemos denotado por $\rho(B)$ el radio espectral de B . Si se hace iteraciones hacia atrás de esta desigualdad obtenemos que

$$\|\vec{e}^{(k+1)}\| \leq [\rho(B)]^k \|\vec{e}^{(0)}\|, \quad k \in \mathbb{N} \quad (2.9)$$

De manera que $\vec{e}^{(k)}$ tiende hacia cero cuando k es suficientemente grande y consecuentemente

hacia $\vec{x}^{(0)}$, siempre que $\rho(B) < 1$. Esta cualidad es una condición necesaria para la convergencia de la sucesión $\vec{x}^{(k)}$. A continuación, damos una condición suficiente y necesaria para garantizar la convergencia de la sucesión $\vec{x}^{(k)}$ hacia \vec{x} .

Proposición 2.1. *Para un método iterativo de la forma (2.7) cuya matriz de iteración, B , satisfice (2.8), se verifica la convergencia para cualquier $\vec{x}^{(0)}$ si y solamente si $\rho(B) < 1$.*

Demostración. A partir del hecho que $\vec{e}^{(k)} = \vec{x} - \vec{x}^{(k)}$, obtenemos la siguiente relación recursiva:

$$\vec{e}^{(k+1)} = B\vec{e}^{(k)}.$$

Dado que $\vec{e}^{(k)} = B^k\vec{e}^{(0)}$, observamos que si $B \in M_{n \times n}(\mathbb{R})$, entonces $\lim_{k \rightarrow \infty} B^k = 0$ si y solo si $\lim_{k \rightarrow \infty} B^k\vec{y} = 0$ para cualquier $\vec{y} \in \mathbb{R}^n$. Esto implica que si $\vec{e}^{(k)} = \vec{0}$, entonces $\lim_{k \rightarrow \infty} B^k\vec{e}^{(0)} = \vec{0}$ en virtud de que $\lim_{k \rightarrow \infty} B^k = 0$.

Ahora, consideremos λ como un valor propio de A con $A^k\vec{x} = \lambda^k\vec{x}$, donde $\vec{x} \neq \vec{0}$ es un vector propio asociado a λ . En este caso, $\lim_{k \rightarrow \infty} \lambda^k = 0$ siempre que $|\lambda| < 1$. Dado que λ puede ser cualquier valor propio de A , concluimos que el radio espectral de B , $\rho(B)$, es menor que 1.

Así, hemos demostrado que $\rho(B) < 1$. ■

Es evidente que a medida que $\rho(B)$ disminuye, se requerirán menos iteraciones para reducir el error inicial en un factor específico.

2.2.1. Construcción de métodos iterativos

La construcción de métodos iterativos se fundamenta en la descomposición matricial de la forma $A = P - (P - A)$, con P una matriz invertible adecuada. A partir de esta relación, se obtiene la expresión:

$$P\vec{x} = (P - A)\vec{x} + \vec{b}$$

cuya estructura es similar a la ecuación (2.8). Mediante la elección de $B = P^{-1}(P - A) = I - P^{-1}A$ y $\vec{g} = P^{-1}\vec{b}$, se plantea el siguiente método iterativo:

$$P(\vec{x}^{(k+1)} - \vec{x}^{(k)}) = \vec{r}^{(k)}, \quad k \geq 0$$

donde

$$\vec{r}^{(k)} = \vec{b} - A\vec{x}^{(k)}, \tag{2.10}$$

representa el residuo vectorial en la k -ésima iteración. Una generalización de este método iterativo se formula como:

$$P(\vec{x}^{(k+1)} - \vec{x}^{(k)}) = \alpha_k \vec{r}^{(k)}, \quad k \geq 0 \quad (2.11)$$

donde $\alpha_k \neq 0$, el cual puede ser ajustado en cada iteración k , con el fin de influir en las propiedades de convergencia de la sucesión $\vec{x}^{(k)}$.

La implementación de este método (2.11) involucra el cálculo del residuo preconditionado $\vec{z}^{(k)}$, obtenido a través de la resolución del sistema lineal correspondiente:

$$P\vec{z}^{(k)} = \vec{r}^{(k)}. \quad (2.12)$$

Con esto, el nuevo valor en la iteración se define como $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \alpha_k \vec{z}^{(k)}$. De ahí la importancia de seleccionar adecuadamente P para lograr una eficiente solución de (2.12), por ejemplo, optando por P diagonal, triangular o tridiagonal.

En las secciones subsiguientes, se aborda tres métodos particulares: Jacobi, Gauss-Seidel y SOR. Cada uno de estos métodos se diferencia por sus características únicas y ventajas en lo que respecta a convergencia y eficiencia, consolidándolos como recursos esenciales para la resolución de problemas relacionados con sistemas lineales.

2.2.2. Método de Jacobi

Si los elementos diagonales de A son no nulos, descomponemos la matriz A como $A = L + D + U$ con L una matriz triangular inferior, U una matriz triangular superior y $D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$ es la matriz diagonal conteniendo los elementos diagonales de A . Entonces, podemos escribir el sistema de ecuaciones $A\vec{x} = \vec{b}$ como sigue:

$$\begin{aligned} A\vec{x} &= (L + D + U)\vec{x} = \vec{b}, \\ D\vec{x} + (L + U)\vec{x} &= \vec{b}, \\ D\vec{x} &= \vec{b} - (L + U)\vec{x}, \end{aligned}$$

que nos da

$$\vec{x} = -D^{-1}(L + U)\vec{x} + D^{-1}\vec{b}.$$

Así, el método de Jacobi corresponde a

$$\vec{x}^{(k+1)} = -D^{-1}(L + U)\vec{x}^{(k)} + D^{-1}\vec{b}. \quad (2.13)$$

Mientras que si vemos a $A = P - (P - A)$ con P una matriz invertible adecuada ($P = D$) obtenemos que

$$\begin{aligned} D\vec{x}^{(k+1)} &= \vec{b} - (A - D)\vec{x}^{(k)}, \quad k \in \mathbb{N}, \\ \vec{x}^{(k+1)} &= -D^{-1}(A - D)\vec{x}^{(k)} + D^{-1}\vec{b}, \quad k \in \mathbb{N} \end{aligned}$$

o, componente a componente,

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)} \right), \quad i = 1, \dots, n \quad (2.14)$$

donde $k \in \mathbb{N}$ y $\vec{x}^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})^T$ es el vector inicial. La matriz de iteración es, por tanto,

$$B = D^{-1}(D - A) = \begin{bmatrix} 0 & -a_{12}/a_{11} & \cdots & -a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & & -a_{2n}/a_{22} \\ \vdots & & \ddots & \vdots \\ -a_{n1}/a_{nn} & -a_{n2}/a_{nn} & \cdots & 0 \end{bmatrix} \quad (2.15)$$

El siguiente resultado nos permitirá verificar más fácilmente la convergencia del método sin necesidad del cálculo de $\rho(B)$.

Proposición 2.2. *Si la matriz A es estrictamente diagonalmente dominante, entonces el método de Jacobi converge.*

Demostración. La demostración de este teorema se encuentra en (Quarteroni et al., 2007, p. 152). ■

2.2.3. Método de Gauss-Seidel

Recordemos que cuando aplicamos el método de Jacobi, cada componente del vector $\vec{x}^{(k)}$, se calcula independientemente de las otras componentes. Parecería que podemos sugerir que podríamos alcanzar una convergencia más rápida si las nuevas componentes $x_j^{(k+1)}, j = 1, \dots, i-1$, se empleasen al mismo tiempo que las componentes $x_j^{(k)}, j \geq i$, para el cómputo de $x_i^{(k+1)}$. Esto nos conduciría a cambiar (2.14) de la siguiente manera:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right), \quad i = 1, \dots, n. \quad (2.16)$$

Este método, denominado método de Gauss-Seidel, concierne a la elección $P = D - L$ y de esta forma podemos escribir el sistema de ecuaciones $A\vec{x} = \vec{b}$ del modo siguiente:

$$\begin{aligned} A\vec{x} &= (D + L + U)\vec{x} = \vec{b} \\ (D + L)\vec{x} + U\vec{x} &= \vec{b} \\ (D + L)\vec{x} &= \vec{b} - U\vec{x} \end{aligned}$$

que nos da

$$\vec{x} = -(D + L)^{-1}U\vec{x} + (D + L)^{-1}\vec{b}.$$

De esta manera, el método de Gauss-Seidel corresponde a

$$\vec{x}^{(k+1)} = -(D + L)^{-1}U\vec{x}^{(k)} + (D + L)^{-1}\vec{b}. \quad (2.17)$$

en tanto que si observamos a $A = P - (P - A)$ con $P = D - L$ tenemos que

$$\begin{aligned} (D - L)\vec{x}^{(k+1)} &= \vec{b} + (D - L - A)\vec{x}^{(k)}, \quad k \in \mathbb{N}, \\ \vec{x}^{(k+1)} &= (D - L)^{-1}(D - L - A)\vec{x}^{(k)} + (D - L)^{-1}\vec{b}, \quad k \in \mathbb{N} \end{aligned}$$

o, componente a componente, obtenemos la expresión (2.17) donde $k \in \mathbb{N}$ y $\vec{x}^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})^T$ es el vector inicial.

Donde la matriz de iteración es, por ende,

$$B = (D - L)^{-1}(D - L - A). \quad (2.18)$$

La proposición que sigue, nos brindará una condición suficiente para garantizar la convergencia del método de Gauss-Seidel.

Proposición 2.3. *Si A es una matriz estrictamente diagonalmente dominante, entonces el método de Gauss-Seidel es convergente.*

Demostración. Notemos que la matriz de iteración del método de Gauss-Seidel, viene dada por $B = (D - L)^{-1}(D - L - A)$, o similarmente, $B = [I - (D - L)^{-1}A]$.

Luego, consideremos un valor propio λ de la matriz B . Así, $[I - (D - L)^{-1}A]\vec{x} = \lambda\vec{x}$, lo que es equivalente a

$$(D + L - A)\vec{x} = \lambda(D + L)\vec{x}.$$

Observemos que

$$\lambda(D+L)\vec{x} = \lambda \sum_{j=1}^i a_{ij}x_j, \quad i = 1, \dots, n$$

y

$$(D+L-A)\vec{x} = - \sum_{j=i+1}^n a_{ij}x_j, \quad i = 1, \dots, n,$$

de estas dos igualdades se sigue que

$$- \sum_{j=i+1}^n a_{ij}x_j = \lambda \sum_{j=1}^i a_{ij}x_j$$

a su vez,

$$\lambda a_{ii}x_i = -\lambda \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}x_j.$$

Tomando valor absoluto (y además por desigualdad triangular) en la igualdad previa, notamos que

$$|\lambda a_{ii}x_i| \leq |\lambda| \sum_{j=1}^{i-1} |a_{ij}| |x_j| + \sum_{j=i+1}^n |a_{ij}| |x_j|.$$

Seguidamente,

$$|\lambda a_{ii}x_i| \leq |\lambda| \sum_{j=1}^{i-1} |a_{ij}| \|x\|_{\infty} + \sum_{j=i+1}^n |a_{ij}| \|x\|_{\infty}.$$

Tomamos en consideración solamente vectores $\vec{x} \neq \vec{0}$ tales que $\|\vec{x}\|_{\infty} = 1$, puesto que podemos definir $\vec{y} := \frac{\vec{x}}{\|\vec{x}\|_{\infty}}$, que será un vector propio de la matriz A . Si tenemos que $\|\vec{x}\|_{\infty} = 1$, entonces existe $i \in \{1, \dots, n\}$ tal que $|x_i| = 1$, y de esta manera

$$|\lambda| \leq \frac{\sum_{j=i+1}^n |a_{ij}|}{|a_{ii}| - \sum_{j=1}^{i-1} |a_{ij}|}. \quad (2.19)$$

Por otra parte, como A es una matriz estrictamente diagonalmente dominante, se observa que

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| = \sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^n |a_{ij}|,$$

de donde vemos que

$$|a_{ii}| - \sum_{j=1}^{i-1} |a_{ij}| > \sum_{j=i+1}^n |a_{ij}| \quad (2.20)$$

Reemplazando (2.20) en (2.19) comprobamos que $|\lambda| < 1$ y por ende $\rho(B) < 1$.

De este modo, por la Proposición 2.1 se sigue la convergencia del método de Gauss-Seidel. ■

Mencionamos también para garantizar la convergencia del método de Gauss-Seidel, existen matrices especiales A cuyas matrices de iteración asociadas cumplen con las hipótesis de la Proposición 2.1; entre las cuales destacamos: matrices que son estrictamente diagonalmente dominantes y matrices que son simétricas y definidas positivas.

2.2.4. Método de Sobrerelajacion Sucesiva

Una posible generalización es el llamado método de relajación en el cual $P = \frac{1}{\omega}D - L$, $\omega \neq 0$ es el parámetro de relajación. En efecto, notemos que a partir de

$$\left(\frac{1}{\omega}D + L\right)\vec{x}^{(k+1)} = -U\vec{x}^{(k)} + \vec{b}$$

se sigue que

$$\vec{x}^{(k+1)} = -\omega D^{-1}L\vec{x}^{(k+1)} - \omega D^{-1}U\vec{x}^{(k)} + \omega D^{-1}\vec{b}.$$

Si sumamos y restamos $\vec{x}^{(k)}$ a ambos lados de la igualdad anterior, obtenemos

$$\begin{aligned}\vec{x}^{(k+1)} - \vec{x}^{(k)} &= \omega D^{-1}(\vec{b} - L\vec{x}^{(k+1)} - U\vec{x}^{(k)} - D\vec{x}^{(k)}) \\ \vec{x}^{(k+1)} &= \vec{x}^{(k)} + \omega D^{-1}(\vec{b} - L\vec{x}^{(k+1)} - U\vec{x}^{(k)} - D\vec{x}^{(k)}).\end{aligned}$$

Pasando al lado izquierdo los términos que tienen $\vec{x}^{(k)}$ se observa que

$$\left(\frac{1}{\omega}D + L\right)\vec{x}^{(k+1)} = \left[\left(\frac{1}{\omega} - 1\right)D - U\right]\vec{x}^{(k)} + \vec{b},$$

y a su vez

$$(D + \omega L)\vec{x}^{(k+1)} = [(1 - \omega)D - \omega U]\vec{x}^{(k)} + \omega\vec{b}.$$

De esta forma, el método SOR o de relajación corresponde a

$$\vec{x}^{(k+1)} = (D + \omega L)^{-1}[(1 - \omega)D - \omega U]\vec{x}^{(k)} + \omega(D + \omega L)^{-1}\vec{b}. \quad (2.21)$$

En este caso la matriz de iteración B es, respectivamente

$$B = (D + \omega L)^{-1}[(1 - \omega)D - \omega U] \quad (2.22)$$

Si $\omega = 1$ este método coincide con el método de Gauss-Seidel. Además, cuando $0 < \omega < 1$ el método se nombra sub-relajación en tanto que si $\omega > 1$ se nombra sobre-relajación.

Tal como en los dos métodos precedentes, podemos ver a este método componente a componente

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right] + (1 - \omega)x_i^{(k)}, \quad i = 1, \dots, n. \quad (2.23)$$

Con el propósito de ilustrar la aplicación práctica del Método de Sobrerrelajación Sucesiva y, simultáneamente, profundizar en la comprensión de su funcionamiento, examinemos un ejemplo concreto.

Ejemplo. Consideremos el siguiente sistema de ecuaciones lineales:

$$\begin{cases} 5x_1 - 7x_2 + 2x_3 = 4, \\ 4x_1 + 6x_2 + 7x_3 = 1, \\ 2x_1 + 2x_2 + 2x_3 = 0, \end{cases}$$

donde $\bar{x}^{(0)} = (0, 0, 0)^\top$ representa la aproximación inicial y $\omega = 1.4$ es el parámetro de relajación.

Las ecuaciones para el método SOR se expresan de la siguiente manera:

$$\begin{aligned} x_1^{(k+1)} &= \frac{\omega}{a_{11}} \left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \right] + (1 - \omega)x_1^{(k)}, \text{ para } i = 1 \\ x_2^{(k+1)} &= \frac{\omega}{a_{22}} \left[b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} \right] + (1 - \omega)x_2^{(k)}, \text{ para } i = 2 \\ x_3^{(k+1)} &= \frac{\omega}{a_{33}} \left[b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} \right] + (1 - \omega)x_3^{(k)}, \text{ para } i = 3 \end{aligned}$$

Las ecuaciones correspondientes para el sistema lineal dado están dadas por:

$$\begin{aligned} x_1^{(k+1)} &= \left(\frac{1.4}{5} \right) \left[4 + 7x_2^{(k)} - 2x_3^{(k)} \right] - 0.4x_1^{(k)} \\ x_2^{(k+1)} &= \left(\frac{1.4}{6} \right) \left[1 - 4x_1^{(k+1)} - 7x_3^{(k)} \right] - 0.4x_2^{(k)} \\ x_3^{(k+1)} &= \left(\frac{1.4}{2} \right) \left[-2x_1^{(k+1)} - 2x_2^{(k+1)} \right] - 0.4x_3^{(k)} \end{aligned}$$

La primera aproximación $\bar{x}^{(1)}$ se obtiene utilizando la aproximación inicial $\bar{x}^{(0)} = (0, 0, 0)^\top$:

$$\begin{aligned} x_1^{(1)} &= \left(\frac{1.4}{5} \right) \left[4 + 7x_2^{(0)} - 2x_3^{(0)} \right] - 0.4x_1^{(0)} \\ &\approx 1.12 \\ x_2^{(1)} &= \left(\frac{1.4}{6} \right) \left[1 - 4x_1^{(1)} - 7x_3^{(0)} \right] - 0.4x_2^{(0)} \\ &\approx -0.812 \end{aligned}$$

$$x_3^{(1)} = \left(\frac{1.4}{2}\right) [-2x_1^{(1)} - 2x_2^{(1)}] - 0.4x_3^{(0)}$$

$$\approx -0.4312$$

La segunda aproximación $\bar{x}^{(2)}$ se determina utilizando la primera aproximación:

$$x_1^{(2)} = \left(\frac{1.4}{5}\right) [4 + 7x_2^{(1)} - 2x_3^{(1)}] - 0.4x_1^{(1)}$$

$$\approx -0.678048$$

$$x_2^{(2)} = \left(\frac{1.4}{6}\right) [1 - 4x_1^{(2)} - 7x_3^{(1)}] - 0.4x_2^{(1)}$$

$$\approx 1.895271467$$

$$x_3^{(2)} = \left(\frac{1.4}{2}\right) [-2x_1^{(2)} - 2x_2^{(2)}] - 0.4x_3^{(1)}$$

$$\approx -1.531632854$$

Por lo tanto, las dos primeras iteraciones están dadas por:

$$\bar{x}^{(1)} \approx (1.12, -0.812, -0.4312)^\top,$$

$$\bar{x}^{(2)} \approx (-0.678048, 1.895271467, -1.531632854)^\top.$$

Finalmente, podemos determinar el error relativo entre las iteraciones

$$\frac{\|\bar{x}^{(2)} - \bar{x}^{(1)}\|_\infty}{\|\bar{x}^{(2)}\|_\infty} \approx \frac{2.707271467}{1.895271467} \approx 1.43.$$

En este ejemplo particular, hemos demostrado la aplicación del método SOR en la resolución de sistemas lineales, destacando su papel esencial en la aceleración de la convergencia hacia la solución deseada mediante la selección adecuada del parámetro de relajación ω , aunque nos centramos en las dos primeras iteraciones en este ejemplo, en la práctica se ejecutan más iteraciones con el fin de alcanzar niveles de precisión superiores, este proceso iterativo, respaldado por la elección cuidadosa de ω , conduce a una convergencia más rápida y eficiente hacia la solución exacta del sistema de ecuaciones lineales.

La convergencia del método SOR es de suma relevancia, dado que asegura que las iteraciones llevarán a una solución deseada. En este contexto, presentamos el siguiente teorema, el cual establece una condición esencial para dicha convergencia.

Teorema 2.4. *Si el método SOR converge, entonces el parámetro ω satisface que $0 < \omega < 2$.*

Demostración. La matriz de iteraciones para el método SOR viene dado por

$$B = (D + \omega L)^{-1}[(1 - \omega)D - \omega U] \quad (2.24)$$

Usando las propiedades del determinante para la matriz B , podemos calcular:

$$\begin{aligned} \det(B) &= \det[(D + \omega L)^{-1}[(1 - \omega)D - \omega U]] \\ &= \det\left[[D(I + \omega D^{-1}L)]^{-1}\right] \det[D[(1 - \omega)I - \omega D^{-1}U]]. \\ &= \det(D^{-1}) \det\left[(I + \omega D^{-1}L)^{-1}\right] \det(D) \det[(1 - \omega)I - \omega D^{-1}U] \\ &= \det[(1 - \omega)I - \omega D^{-1}U] \\ &= (1 - \omega)^n, \end{aligned}$$

donde n denota la dimensión de la matriz A . También conocemos que el determinante de una matriz es igual al producto de sus valores propios. Por ende, si consideremos que

$$\det(B) = \prod_{i=1}^n \lambda_i$$

con λ_i valor propio de B , entonces seguimos que

$$\prod_{i=1}^n \lambda_i = (1 - \omega)^n.$$

Adicionalmente el radio espectral tiene la siguiente relación con los valores propios λ_i

$$\rho(B) \geq |\lambda_i|, \quad \text{con } i = 1, \dots, n,$$

de donde se desprende que

$$|1 - \omega|^n = \prod_{i=1}^n |\lambda_i| \leq \prod_{i=1}^n \rho(B) = [\rho(B)]^n$$

y así arribamos a

$$|1 - \omega| \leq \rho(B).$$

Ahora sabemos que para que el método iterativo converja debemos garantizar que $\rho(B) < 1$ y de este hecho se desprende que

$$|1 - \omega| < 1 \iff 0 < \omega < 2.$$

■

El siguiente resultado desempeña un papel fundamental en la comprensión de las propiedades de convergencia de los métodos iterativos empleados para resolver sistemas de ecuaciones lineales asociados con matrices tridiagonales simétricas. Además, establece una conexión significativa entre los métodos de Jacobi y SOR en este contexto específico.

Teorema 2.5. Sea $A = D - L - U$ una matriz que satisface la condición siguiente

$$\det(kD - \gamma L - \gamma^{-1}U) = \det(kD - L - U) \quad \text{para todo } k, \gamma \in \mathbb{R} \setminus \{0\} \quad (2.25)$$

y B_J y B_ω las matrices de iteración de los métodos de Jacobi y SOR, como se definió anteriormente. Si μ es un valor propio de B_J y $\lambda \neq 0$ satisface

$$\mu = \frac{\lambda + \omega - 1}{\omega \lambda^{1/2}} \quad (2.26)$$

para algun $\omega \in (0, 2)$, entonces λ es un valor propio de B_ω .

Demostración. Las matrices de iteraciones B_J y B_ω vienen dado por

$$B_J = D^{-1}(L + U)$$

$$B_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U].$$

Consideremos $\lambda \neq 0$ un valor propio de la matriz B_ω . Como L y U son matrices triangulares superior e inferior respectivamente, se tiene que $\det(D^{-1}) \det(D - \omega L) = 1$.

$$\begin{aligned} \det(B_\omega - \lambda I) &= \det[(D - \omega L)^{-1}[(1 - \omega)D + \omega U] - \lambda I] \\ &= \det(D^{-1}) \det(D - \omega L) \det[(D - \omega L)^{-1}[(1 - \omega)D + \omega U] - \lambda I] \\ &= \det[(1 - \omega)I + \omega D^{-1}U - \lambda I D^{-1}(D - \omega L)] \\ &= \det[(1 - \omega)I + \omega D^{-1}U - \lambda I + \omega \lambda D^{-1}L]. \end{aligned}$$

Luego, sacando factor común ω , y puesto que la matriz de iteración es de dimensión de $n \times n$ se tiene:

$$= \omega^n \det \left[\left(\frac{1}{\omega} - 1 \right) I + D^{-1}U - \frac{\lambda I}{\omega} + \lambda D^{-1}L \right],$$

agrupando términos semejantes, se sigue que:

$$\begin{aligned}
&= \omega^n \det \left[\left(\frac{1}{\omega} - 1 - \frac{\lambda}{\omega} \right) I + D^{-1}U + \lambda D^{-1}U \right] \\
&= \omega^n \det \left[\left(1 + \frac{\lambda}{\omega} - \frac{1}{\omega} \right) (-I) - D^{-1}(U + \lambda D)(-I) \right] \\
&= (-1)^n \omega^n \det(D^{-1}) \det \left[\left(\frac{\omega + \lambda - 1}{\omega} \right) D - (U + \lambda D) \right] \\
&= (-1)^n \omega^n \det(D^{-1}) \det \left[\left(\frac{\omega + \lambda - 1}{\omega} \right) D - U - \lambda D \right] \\
&= \lambda^{1/2} (-1)^n \omega^n \det(D^{-1}) \det \left[\left(\frac{\omega + \lambda - 1}{\omega \lambda^{1/2}} \right) D - \lambda^{-1/2}U - \lambda^{1/2}L \right].
\end{aligned}$$

Aplicando la condición dada en (2.25) tenemos

$$\begin{aligned}
&= (-1)^n \omega^n \lambda^{1/2} \det(D^{-1}) \det \left[\left(\frac{w + \lambda - 1}{w \lambda^{1/2}} \right) D - U - L \right] \\
&= (-1)^n \omega^n \lambda^{1/2} \det \left[\left(\frac{w + \lambda - 1}{w \lambda^{1/2}} \right) I - D^{-1}U - D^{-1}L \right] \\
&= (-1)^n \omega^n \lambda^{1/2} \det \left[\left(\frac{w + \lambda - 1}{\omega \lambda^{1/2}} \right) I - D^{-1}(U + L) \right] \\
&= \omega^n \lambda^{1/2} \det \left[B_j - \left(\frac{\omega + \lambda - 1}{\omega \lambda^{1/2}} \right) \right]
\end{aligned}$$

Puesto que $\lambda \neq 0$ satisface

$$\mu = \frac{\lambda + \omega - 1}{\omega \lambda^{1/2}}.$$

donde μ es un valor propio de B_j . A su vez, cuando μ es un valor propio de B_j , y λ satisface esa relación, entonces λ es un valor propio de B_ω . ■

Las deducciones y fundamentos expuestos en esta sección serán esenciales para comprender y aplicar con éxito el método de Sobrerrelajación Sucesiva (SOR) en el contexto de los esquemas de diferencias finitas utilizados en la aproximación numérica de la ecuación de Poisson. Estas bases teóricas nos capacitarán para examinar en detalle el comportamiento del parámetro de relajación óptimo, su impacto en la convergencia y la eficacia de la simulación numérica.

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo, exploramos los fundamentos que sustentan el enfoque metodológico aplicado en esta investigación, el estudio que se llevó a cabo se enmarca en una metodología cualitativa con un nivel descriptivo, centrada principalmente en la revisión exhaustiva de fuentes secundarias; el objetivo principal de esta fase fue establecer los cimientos necesarios para comprender el funcionamiento del método SOR, posteriormente, este método fue implementado en el lenguaje de programación Python con el propósito de evaluar su eficacia y validar su aplicabilidad.

3.1. Descripción de enfoque, alcance, diseño, tipo, métodos, técnicas e instrumentos de investigación empleadas

En este estudio, adoptaremos un enfoque cualitativo con el objetivo de profundizar en los fundamentos teóricos subyacentes al método SOR, buscando deducir su criterio de convergencia y determinar el valor óptimo del parámetro de relajación aplicado a un esquema de diferencias finitas asociado a la ecuación de Poisson.

Este enfoque se complementa con un alcance de investigación descriptivo que abarca los métodos iterativos, con énfasis en el método SOR como herramienta para la resolución numérica de la ecuación de Poisson, y se enfoca en el análisis destinado a optimizar el parámetro de relajación ω con el objetivo de acelerar la convergencia del método.

Este estudio se ha llevado a cabo bajo un diseño de tipo documental, donde se recopiló información existente mediante una revisión selectiva de documentos relacionados con la optimización del parámetro del método SOR aplicado al sistema de ecuaciones provenientes del método de diferencias finitas para la ecuación de Poisson. Para la implementación práctica de estos conceptos, hemos utilizado el lenguaje de programación Python, conocido por su sintaxis amigable, naturaleza de código abierto, capacidad para generar gráficos claros y eficiencia en el tiempo de ejecución, estas cualidades han convertido a Python en una elección exitosa en diversas aplicaciones de Análisis Numérico.

CAPÍTULO IV

4. MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

En este capítulo, se procede a detallar el proceso iterativo del método SOR. Se describe la actualización de las variables desconocidas en cada iteración, utilizando una combinación de los valores anteriores y los valores de las iteraciones anteriores. Se presentan las expresiones matemáticas involucradas en la actualización y se esclarece el parámetro de sobrerelajación, ω , que controla la rapidez de convergencia del método. Se estudian las ventajas de la sobrerelajación en términos de acelerar la convergencia en comparación con los métodos de Gauss-Seidel y Jacobi.

Luego, se aborda el análisis de la convergencia del método SOR. Se presentan criterios de convergencia y se discute conforme la elección del parámetro de sobrerelajación puede afectar la velocidad de convergencia y la estabilidad del método. Se muestran ejemplos tomados de la literatura que ilustran cómo el método SOR puede converger más rápidamente en ciertos casos en comparación con otros métodos iterativos.

La información que se presenta a continuación se ha extraído de la siguiente fuente académica: (Yang y Matthias, 2007).

4.1. Parámetro óptimo de relajación del Método SOR.

En el contexto del análisis y la optimización del parámetro ω del método SOR, exploraremos la importancia de elegir un valor óptimo para el parámetro de relajación, examinando cómo afecta la convergencia, la estabilidad y la eficiencia del método en la resolución de sistemas de ecuaciones lineales. El siguiente teorema establece condiciones para la convergencia y proporciona una fórmula para optimizar el parámetro de relajación del método SOR.

Teorema 4.1. *Consideremos la matriz A definida como $A = D - L - U$, donde A satisface la ecuación (2.25), y supongamos que $B_J = I - D^{-1}A$ tiene valores propios reales y que $\beta \equiv \rho(B_J) < 1$. Entonces, para cada $\omega \in]0, 2[$, la iteración SOR converge, y radio espectral de la matriz SOR es:*

$$\rho(B_\omega) = \begin{cases} \frac{1}{4} \left[\omega\beta + \sqrt{(\omega\beta)^2 - 4(\omega - 1)} \right]^2, & 0 < \omega \leq \omega_{opt} \\ \omega - 1, & \omega_{opt} \leq \omega < 2 \end{cases} \quad (4.1)$$

donde ω_{opt} , es el valor óptimo de ω , el cual viene dado por:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \beta^2}}. \quad (4.2)$$

Demostración. Resolvemos la ecuación (2.26) para λ de la siguiente manera:

$$\lambda = \frac{1}{4} \left(\mu\omega \pm \sqrt{\mu^2\omega^2 - 4(\omega - 1)} \right)^2,$$

donde μ es un valor propio de la matriz de iteración B_J , de acuerdo con el Teorema 2.5, sabemos que los valores propios de la matriz de iteración B_J son valores propios de la matriz de iteración B_ω . Ahora, consideremos dos casos:

Caso 1: Supongamos que $\mu^2\omega^2 - 4(\omega - 1) < 0$. En este caso, λ será negativo si satisface la siguiente condición:

$$\frac{2 - 2\sqrt{1 - \mu^2}}{\mu^2} < \omega < \frac{2 + 2\sqrt{1 - \mu^2}}{\mu^2},$$

cabe notar que para que el método iterativo SOR converja, debemos garantizar que $0 < \omega < 2$, lo que nos lleva a la siguiente desigualdad:

$$\tilde{\omega} = \frac{2 - 2\sqrt{1 - \mu^2}}{\mu^2} < \omega < 2. \quad (4.3)$$

Determinemos los valores de ω para los cuales λ será negativo:

$$\begin{aligned} \lambda^{\frac{1}{2}} &= \frac{1}{2} \left(\mu\omega \pm i\sqrt{\mu^2\omega^2 - 4(\omega - 1)} \right) \\ |\lambda^{\frac{1}{2}}| &= \sqrt{\frac{1}{4}\mu^2\omega^2 + \frac{1}{4}i^2[\mu^2\omega^2 - 4(\omega - 1)]} \\ &= \sqrt{\frac{1}{4}[\mu^2\omega^2 - \mu^2\omega^2 + 4(\omega - 1)]} \\ &= \sqrt{\frac{4(\omega - 1)}{4}} \\ &= \sqrt{\omega - 1}. \end{aligned}$$

Luego, obtenemos:

$$|\lambda| = \omega - 1, \quad \omega \in [\tilde{\omega}, 2]. \quad (4.4)$$

La expresión (4.4) representa a una función estrictamente creciente cuando $\omega \in (\tilde{\omega}, 2)$ y alcanza su valor mínimo cuando se aproxime a $\tilde{\omega}$.

Caso 2. Consideremos si $\mu^2\omega^2 - 4(\omega - 1) > 0$, lo que implica que $\mu^2\omega^2 > 4(\omega - 1)$. Ahora,

recordemos que las raíces para λ vienen dadas por:

$$\lambda = \frac{1}{4} \left(\omega\mu \pm \sqrt{(\omega\mu)^2 - 4(\omega - 1)} \right)^2,$$

y el valor máximo de λ se expresa como:

$$\lambda_{\max} = \frac{1}{4} \left(\omega\mu + \sqrt{(\omega\mu)^2 - 4(\omega - 1)} \right)^2, \quad (4.5)$$

obteniéndose así lo siguiente:

$$\lambda = \frac{1}{4} \left(\omega\mu + \sqrt{\omega^2\mu^2 - 4(\omega - 1)} \right)^2 \geq \omega^2\mu^2 > 4(\omega - 1) > \omega - 1,$$

de este modo, a partir de la expresión (2.26) del Teorema 2.5, se sigue que

$$\mu = \frac{\lambda + \omega - 1}{\omega\lambda^{\frac{1}{2}}} \geq \frac{(\omega - 1) + (\omega - 1)}{\omega(\omega - 1)^{\frac{1}{2}}} = \frac{2(\omega - 1)}{\omega(\omega - 1)^{\frac{1}{2}}} = \frac{2(\omega - 1)^{\frac{1}{2}}}{\omega} \geq 0.$$

Luego, por definición de valor absoluto, notemos que $|\mu| = \mu$ si $\mu \geq 0$, y considerando la premisa inicial de que $\beta \equiv \rho(B_J) < 1$, y que la matriz de iteración B_J tiene valores propios reales, garantizamos que μ es un número real. Por lo tanto, podemos utilizar esta propiedad en la expresión (4.5), obteniendo así:

$$\lambda_{\max} = \frac{1}{4} \left[\omega|\mu| + \sqrt{(\omega|\mu|)^2 - 4(\omega - 1)} \right]^2.$$

En efecto, para determinar la monotonía de λ_{\max} según el criterio de la derivada, denotada como $\lambda'_{\max}(|\mu|)$, que representa la derivada de λ_{\max} respecto a $|\mu|$, procedemos de la siguiente manera:

$$\lambda'_{\max}(|\mu|) = \frac{1}{2} \left(\omega|\mu| + \sqrt{(\omega|\mu|)^2 - 4(\omega - 1)} \right) \left(|\mu| + \frac{|\mu|^2\omega - 2}{\sqrt{(\omega|\mu|)^2 - 4(\omega - 1)}} \right).$$

Considerando $|\mu| = \rho(B_J) = \beta$, como se establece en el Teorema 2.5, se sigue que β es un valor propio de B_ω y es equivalente al valor propio de B_J , consecuentemente

$$\lambda'_{\max}(\omega) = \frac{1}{2} \underbrace{\left(\omega\beta + \sqrt{(\omega\beta)^2 - 4(\omega - 1)} \right)}_{\geq 0} \left(\beta + \frac{\beta^2\omega - 2}{\sqrt{(\omega\beta)^2 - 4(\omega - 1)}} \right).$$

Para determinar el signo de λ'_{\max} , examinemos el signo de la expresión

$$\beta + \frac{\beta^2\omega - 2}{\sqrt{(\omega\beta)^2 - 4(\omega - 1)}},$$

de la cual se nota que $\sqrt{(\omega\beta)^2 - 4(\omega - 1)} > 0$, $\beta < 1$, $\omega < 2$, advirtiéndose que

$$\beta \sqrt{(\omega\beta)^2 - 4(\omega - 1)} + \omega\beta^2 - 2 < \sqrt{\omega^2 - 4\omega + 4} + \omega - 2 = \sqrt{(\omega - 2)^2} + \omega - 2 = 0.$$

del cual, $\lambda'_{\text{máx}}(\omega) < 0$ sobre $]0, \tilde{\omega}]$, es decir que $\lambda_{\text{máx}}$ es una función decreciente en este intervalo. Consecuentemente, el mínimo se alcanza en $\tilde{\omega}$ definido en (4.3) tal como se muestra a continuación:

$$\tilde{\omega} = \frac{2(1 - \sqrt{1 - |\mu|^2})}{|\mu|^2} = \frac{2(1 - \sqrt{1 - \beta^2})}{\beta^2} \cdot \frac{1 + \sqrt{1 - \beta^2}}{1 + \sqrt{1 - \beta^2}} = \frac{2(1 - (1 - \beta^2))}{\beta^2(1 + \sqrt{1 - \beta^2})},$$

en adición,

$$= \frac{2(1 - 1 + \beta^2)}{\beta^2(1 + \sqrt{1 - \beta^2})} = \frac{2\beta^2}{\beta^2(1 + \sqrt{1 - \beta^2})} = \frac{2}{1 + \sqrt{1 - \beta^2}}$$

y por ende

$$\omega_{\text{opt}} := \frac{2}{1 + \sqrt{1 - \beta^2}}.$$

■

El teorema previamente enunciado proporciona una sólida base teórica para comprender la convergencia y el radio espectral de la matriz de iteración del método SOR, lo cual resulta esencial al considerar la determinación del valor óptimo de ω , denotado como ω_{opt} , que optimiza la velocidad de convergencia de este método. Posteriormente, procederemos a aplicar estos principios en el contexto concreto de la solución numérica de un sistema de ecuaciones originado en la discretización mediante diferencias finitas de la ecuación de Poisson con condiciones de contorno de Dirichlet.

4.2. Discretización de la ecuación de Poisson

En el ámbito de la resolución numérica de problemas de ecuaciones diferenciales, es fundamental comprender la importancia de las aproximaciones numéricas, ya que las soluciones exactas suelen ser difíciles o imposibles de obtener de forma analítica debido a la complejidad y diversidad de los problemas en ingeniería, ciencia y matemáticas. Estas aproximaciones son esenciales para traducir problemas continuos en discretos, lo que permite el uso de algoritmos y técnicas computacionales para obtener soluciones aproximadas que son esenciales en la modelización y simulación de fenómenos en la ciencia y la ingeniería, como en el caso de la resolución numérica de la ecuación de Poisson bajo condiciones de frontera de Dirichlet.

Posteriormente, abordamos la discretización en diferencias finitas, una técnica que se basa en la aproximación de derivadas parciales en ecuaciones diferenciales, lo que conduce a un sistema de ecuaciones lineales, seguidamente consideramos la ecuación de Poisson con las condiciones de frontera de Dirichlet homogéneas:

$$\begin{cases} -u''(x) = f(x), & \text{sobre } x \in]0, 1[, \\ u(1) = u(0) = 0, \end{cases} \quad (4.6)$$

con f una función dada (que se supone tiene cierta regularidad) definida sobre $]0, 1[$. Por ende, el problema es encontrar una función $u(x)$ que satisfaga estas condiciones establecidas en (4.6). Al introducir la aproximación de segundo orden mediante el desarrollo de Taylor, arribamos al siguiente problema discreto:

$$\begin{cases} -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f(x_i), & i \in \{1, \dots, n\} \\ u_0 = 0, u_{n+1} = 0. \end{cases} \quad (4.7)$$

Esta discretización (4.7) se reduce al sistema de ecuaciones lineales:

$$A\vec{U} = \vec{F}. \quad (4.8)$$

Con A la matriz de $M_{n \times n}(\mathbb{R})$ dada por:

$$A = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}, \quad (4.9)$$

mientras que los vectores \vec{U} y \vec{F} son vectores de \mathbb{R}^n definidos por

$$\vec{U} = [u_1, u_2, \dots, u_n]^T \quad \text{y} \quad \vec{F} = h^2 [f(x_1), f(x_2), \dots, f(x_n)]^T \quad (4.10)$$

donde \vec{U} representa el vector de las incógnitas del sistema.

Teóricamente, la solución \vec{U} del sistema lineal (4.9) está obviamente definida por $\vec{U} = A^{-1}\vec{F}$ pero en la práctica no calculamos la inversa de una matriz. Ciertamente, existen múltiples métodos numéricos que permiten resolver sistemas lineales sin requerir el cálculo explícito de la inversa de la matriz. En este contexto, se presenta el método SOR (Successive Over-Relaxation) previamente mencionado, con la

necesidad de realizar una previa verificación de la existencia y unicidad de la solución del sistema (4.9). Esto se realiza teniendo en cuenta que la matriz A es simétrica y tridiagonal. En consecuencia, presentamos el siguiente teorema, que establece las condiciones necesarias y suficientes para garantizar la existencia y unicidad de la solución en el sistema discreto que hemos abordado.

Teorema 4.2. *El problema discreto (4.7) admite solución única si la matriz A definida en (4.9) es simétrica y definida positiva.*

Demostración. Claramente A es simétrica. Ahora procedamos a demostrar que la matriz A es definitivamente positiva. Para ello, consideremos un vector $\vec{x} \in \mathbb{R}^n$ no nulo.

$$\begin{aligned} \vec{x}^T A \vec{x} &= \begin{bmatrix} \vec{x}_1 & \dots & \vec{x}_n \end{bmatrix} \begin{bmatrix} 2\vec{x}_1 - \vec{x}_2 \\ \vdots \\ -\vec{x}_{i-1} + 2\vec{x}_i - \vec{x}_{i+1} \\ \vdots \\ -\vec{x}_{n-1} + 2\vec{x}_n \end{bmatrix} = \left(2\vec{x}_1^2 - \vec{x}_1\vec{x}_2 + \sum_{i=2}^{n-1} \vec{x}_i (-\vec{x}_{i-1} + 2\vec{x}_i - \vec{x}_{i+1}) - \vec{x}_{n-1}\vec{x}_n + 2\vec{x}_n^2 \right) \\ &= \left(2\vec{x}_1^2 - \vec{x}_1\vec{x}_2 + \sum_{i=2}^{n-1} \vec{x}_i (\vec{x}_i - \vec{x}_{i-1}) - \sum_{i=2}^{n-1} \vec{x}_i (\vec{x}_{i+1} - \vec{x}_i) - \vec{x}_{n-1}\vec{x}_n + 2\vec{x}_n^2 \right) \\ &= \left(2\vec{x}_1^2 - \vec{x}_1\vec{x}_2 + \sum_{i=1}^{n-2} \vec{x}_{i+1} (\vec{x}_{i+1} - \vec{x}_i) - \sum_{i=2}^n \vec{x}_i (\vec{x}_{i+1} - \vec{x}_i) - \vec{x}_{n-1}\vec{x}_n + 2\vec{x}_n^2 \right), \end{aligned}$$

notando que

$$\vec{x}^T A \vec{x} = \left(\vec{x}_1^2 + \sum_{i=1}^{n-1} (\vec{x}_{i+1} - \vec{x}_i)^2 + \vec{x}_n^2 \right) \geq 0$$

lo que nos indica que A es simétrica y definida positiva, y a su vez garantiza la existencia y unicidad de la solución para el problema (4.7). ■

Los sistemas de ecuaciones surgen de la discretización de ecuaciones diferenciales, en particular, el sistema de ecuaciones dado en (4.7) es una discretización de la ecuación diferencial de Poisson, así, esta relación es fundamental, pues vincula la aproximación y resolución de sistemas de ecuaciones discretos con la de ecuaciones diferenciales subyacentes. A continuación, presentamos un resultado esencial en la resolución de ecuaciones diferenciales mediante el análisis de las raíces del polinomio característico, en el cual se establece que si todas las raíces de dicho polinomio son únicas y distintas de cero, entonces cada solución de la ecuación diferencial puede expresarse como una combinación lineal de esa solución específica.

Teorema 4.3. *Si todas las raíces del polinomio característico de una ecuación diferencial lineal son simples y distintas de cero, entonces cada solución de la ecuación diferencial es una combinación lineal de esa solución particular.*

En otras palabras, si r_1, r_2, \dots, r_m son raíces simples y distintas de cero para el polinomio característico de una ecuación diferencial, entonces cada componente x_i del vector solución \vec{x} de la ecuación puede ser

expresado como:

$$x_i = \sum_{k=1}^m a_k r_k^i, \quad 1 \leq i \leq m,$$

donde a_k son coeficientes determinados por las condiciones de contorno o condiciones iniciales de la ecuación diferencial.

Demostración. Sea A un operador de desplazamiento definido por $(Ax)_n = x_{n+1}$ y consideremos $u = \{r^1, r^2, r^3, \dots, r^k\}$ donde r^i con $i = 1, \dots, k$ es la solución de la ecuación diferencial parcial. Luego, observemos que:

$$(Au)_n = u_{n+1} = r^{n+1} = r^1 r^n = r^1 u,$$

y a su vez

$$A^2 u = (AAu)_n = u_{n+2} = r^{n+2} = r^2 r^n = r^2 u.$$

De forma general se cumple

$$A^i u = r^i u.$$

Por lo tanto, si p es un polinomio definido por $p(r) = \sum_{i=0}^m c_i r^i$, entonces:

$$p(A)u = \left(\sum_{i=0}^m c_i A^i \right) u = \sum_{i=0}^m c_i (A^i u) = \sum_{i=0}^m (c_i r^i) u = p(r)u,$$

si $p(r) = 0$, entonces $P(A)u = 0$ y así $r_1, r_2, r_3, \dots, r_n$ son raíces del polinomio $p(r)$ correspondiente a la solución de la ecuación diferencial $P(A)x = 0$, es decir, la solución viene dada por $u^{(k)} = (r_k^1, r_k^2, r_k^3, \dots, r_k^n)$.

Supongamos que x es una solución arbitraria de la ecuación diferencial $P(A)x = 0$. Como r_k es una raíz de p , tenemos que $p(r_k) = 0$, lo que implica que $p(A)u^{(k)} = 0$ para todo k .

Por lo tanto, se puede escribir como:

$$x_i = \sum_{k=1}^m a_k u^k$$

$$x_i = \sum_{k=1}^m a_k r_k^i, \quad 1 \leq i \leq m.$$

■

El siguiente teorema presenta una perspicaz alternativa para verificar que una matriz sea definida positiva al calcular sus valores propios, lo cual es fundamental, ya que esta propiedad garantiza la existencia de una única solución para el sistema de ecuaciones correspondiente.

Teorema 4.4. Dada la matriz A definida en la ecuación (4.9), los valores propios λ_k y los vectores propios $z^{(k)}$ de A , con $k = 1, \dots, n$, se expresan como sigue:

$$\lambda_k = 2 \left(1 - \cos \left(\frac{k\pi}{n+1} \right) \right) = 4 \operatorname{sen}^2 \left(\frac{k\pi}{2(n+1)} \right), \quad (4.11)$$

y las componentes de $z^{(k)} = (z_\ell^{(k)})$ se definen como:

$$z_\ell^{(k)} = \sqrt{\frac{2}{n+1}} \operatorname{sen}\left(\frac{k\ell\pi}{n+1}\right), \quad \ell = 1, 2, \dots, n. \quad (4.12)$$

Demostración. Definimos λ_k y $z^{(k)}$ como los valores y vectores propios de la matriz A . Para encontrarlos, consideramos la ecuación característica $\det(A - \lambda_k I) = 0$, que nos conduce a una ecuación diferencial de segundo orden:

$$z_{k+2} - (2 - \lambda_k)z_{k+1} + z_k = 0, \quad k = 1, \dots, m,$$

con condiciones de frontera $z_0 = z_{n+1} = 0$.

El polinomio característico se obtiene al sustituir $z_k = r^k$ en la ecuación anterior, y después de simplificar, llegamos a:

$$r^2 - (2 - \lambda_k)r + 1 = 0. \quad (4.13)$$

El Teorema 4.3 garantiza que la solución general de la ecuación (4.13) es:

$$z_\ell = \alpha r_1^\ell + \beta \ell r_1^\ell,$$

donde α y β son constantes, luego las condiciones de frontera $z_0 = z_{n+1} = 0$ nos conducen al sistema de ecuaciones:

$$\begin{cases} 0 = \alpha + \beta \\ 0 = \alpha r_1^{n+1} + \beta r_2^{n+1} \end{cases}$$

Al resolver este sistema, obtenemos:

$$\left(\frac{r_1}{r_2}\right)^{n+1} = 1 = e^{2\pi ki}, \quad k = 0, \dots, n.$$

Lo que implica:

$$\frac{r_1}{r_2} = e^{\frac{2\pi ki}{n+1}}, \quad k = 0, \dots, n.$$

Entonces, podemos expresar r_1 como:

$$r_1 = r_2 e^{\frac{i2\pi k}{n+1}}.$$

Dado que r_1 y r_2 son las raíces del polinomio característico, podemos escribir:

$$\begin{aligned} p(r) &= r^2 - (2 - \lambda_k)r + 1 \\ &= (r - r_1)(r - r_2) \\ &= r^2 - (r_1 + r_2)r + r_1 r_2 \end{aligned} \quad (4.14)$$

Al igualar los términos lineales de (4.14), obtenemos:

$$2 - \lambda_k = r_1 + r_2.$$

Esto nos lleva a la determinación de los valores propios λ_k , $k = 1, \dots, n$, como:

$$\lambda_k = 2 - (r_1 + r_2) = 2 - \left(e^{\frac{ik\pi}{n+1}} + e^{-\frac{ik\pi}{n+1}} \right) = 2 - 2 \cos \left(\frac{k\pi}{n+1} \right).$$

Por medio de la identidad trigonométrica del ángulo mitad se sigue que

$$4 \operatorname{sen}^2 \left(\frac{\theta}{2} \right) = 2 - 2 \cos(\theta),$$

obtenemos una forma alternativa para los valores propios:

$$\lambda_k = 4 \operatorname{sen}^2 \left(\frac{k\pi}{2(n+1)} \right).$$

Los componentes de los vectores propios se expresan como sigue:

$$z_\ell^{(k)} = \alpha r_1^\ell + \beta r_2^\ell = \alpha (r_1^\ell - r_2^\ell) = \alpha \left(e^{\frac{k\ell\pi i}{n+1}} - e^{-\frac{k\ell\pi i}{n+1}} \right) = 2i\alpha \operatorname{sen} \left(\frac{k\ell\pi}{n+1} \right),$$

donde α se define como $\alpha = \frac{\sqrt{\frac{2}{n+1}}}{2i}$. Esto conduce a la siguiente expresión:

$$z_\ell^{(k)} = \sqrt{\frac{2}{n+1}} \operatorname{sen} \left(\frac{k\ell\pi}{n+1} \right) \quad k = 1, 2, \dots, n.$$

■

A continuación, presentamos una generalización para el cálculo de valores y vectores propios en matrices tridiagonales simétricas, con el objetivo de establecer una conexión relevante con el estudio del método SOR, ya que analiza las propiedades de los valores propios, que determinan la existencia de soluciones en el sistema asociado.

Teorema 4.5. Consideremos la matriz $A \in M_{n \times n}(\mathbb{R})$ definida como:

$$A = \begin{bmatrix} \alpha & \beta & & & \\ \beta & \alpha & \beta & & \\ & \ddots & \ddots & \ddots & \\ & & \beta & \alpha & \beta \\ & & & \beta & \alpha \end{bmatrix}.$$

donde α y β son constantes reales. Entonces, los valores propios λ_k y los vectores propios $z^{(k)}$, para $k = 1, \dots, n$, de la matriz A se definen de la siguiente manera:

$$\lambda_k = \alpha + 2\beta \cos \left(\frac{k\pi}{n+1} \right) \tag{4.15}$$

y el componente m -ésimo de $z^{(k)} = (z_m^{(k)})$ se expresa como:

$$z_m^{(k)} = \sqrt{\frac{2}{n+1}} \operatorname{sen}\left(\frac{mk\pi}{n+1}\right), \quad k, m = 1, 2, \dots, n. \quad (4.16)$$

Demostración. Definimos λ_k y $z^{(k)}$ como los valores y vectores propios de la matriz A . Para encontrarlos, consideramos la ecuación característica $\det(A - \lambda_k I) = 0$, que nos conduce a una ecuación diferencial de segundo orden:

$$\beta z_{k+2} + (\alpha - \lambda) z_{k+1} + \beta z_k = 0 \quad k = 1, \dots, n,$$

donde las condiciones de frontera establecidas son $z_0 = z_{n+1} = 0$.

El polinomio característico se obtiene al sustituir $z_k = r^k$ en la ecuación anterior, y después de simplificar, llegamos a:

$$\beta r^2 - (\alpha - \lambda)r + \beta = 0, \quad (4.17)$$

Según el Teorema 4.3, la solución general de la ecuación (4.17) se expresa como:

$$z_\ell = \alpha r_1^\ell + \beta \ell r_1^\ell,$$

donde α y β son constantes. Las condiciones de frontera $z_0 = z_{n+1} = 0$ nos llevan a resolver el sistema de ecuaciones siguiente:

$$\begin{cases} 0 = \alpha + \beta \\ 0 = \alpha r_1^{n+1} + \beta r_2^{n+1} \end{cases}$$

Al resolver este sistema, obtenemos:

$$\left(\frac{r_1}{r_2}\right)^{n+1} = 1 = e^{2\pi ki}, \quad k = 0, \dots, n.$$

Esto implica:

$$\frac{r_1}{r_2} = e^{\frac{2\pi ki}{n+1}}, \quad k = 0, \dots, n.$$

Entonces, podemos expresar r_1 como:

$$r_1 = r_2 e^{\frac{2\pi k}{n+1}}.$$

Dado que r_1 y r_2 son las raíces del polinomio característico, podemos escribir:

$$\begin{aligned} p(r) &= \beta r^2 - (2 - \lambda_k)r + \beta \\ &= (r - r_1)(r - r_2) \\ &= r^2 - (r_1 + r_2)r + r_1 r_2. \end{aligned} \quad (4.18)$$

Dada la relación $\beta = r_1 r_2$, expresamos r_1 y r_2 como sigue:

$$r_1 = \beta e^{\frac{ik\pi}{n+1}}, \quad r_2 = \beta e^{-\frac{ik\pi}{n+1}}, \quad k = 1, \dots, n.$$

Al igualar los términos lineales en (4.18), obtenemos:

$$(\alpha - \lambda_k) = -(r_1 + r_2).$$

Esto nos conduce a la determinación de los valores propios $\lambda_k, k = 1, \dots, n$, de la siguiente manera:

$$\lambda_k = \alpha + (r_1 + r_2) = \alpha + \beta \left(e^{\frac{ik\pi}{n+1}} + e^{-\frac{ik\pi}{n+1}} \right) = \alpha + 2\beta \cos\left(\frac{k\pi}{n+1}\right).$$

Los componentes de los vectores propios se expresan como sigue:

$$z_m^{(k)} = \alpha r_1^m + \beta r_2^m = \alpha (r_1^m - r_2^m) = \alpha \beta \left(e^{\frac{mk\pi i}{n+1}} - e^{-\frac{mk\pi i}{n+1}} \right) = 2i\alpha\beta \operatorname{sen}\left(\frac{mk\pi}{n+1}\right),$$

donde α se define como $\alpha = \frac{\sqrt{\frac{2}{n+1}}}{2i\beta}$. Esto nos lleva a la siguiente expresión:

$$z_m^{(k)} = \frac{2}{n+1} \operatorname{sen}\left(\frac{mk\pi}{n+1}\right), \quad k, m = 1, 2, \dots, n.$$

■

Los resultados previos, en particular, la verificación de la existencia y unicidad de soluciones en el sistema derivado de la discretización en diferencias finitas de la ecuación de Poisson, son fundamentales; por consiguiente, nos adentramos en la sección siguiente, donde, basados en estos fundamentos teóricos y respaldados por el cálculo de valores y vectores propios en matrices tridiagonales simétricas, nos proporcionan las bases esenciales para abordar la resolución numérica de esta ecuación mediante métodos iterativos.

4.3. Simulación numérica de la ecuación de Poisson.

En esta sección, se presentan las gráficas obtenidas a partir de la simulación numérica para la ecuación de Poisson, aplicada a un esquema de diferencias finitas, lo que nos conduce a resolver un sistema de ecuaciones lineales. El objetivo es determinar el parámetro óptimo de relajación para el método SOR y comparar los resultados con los métodos iterativos de Jacobi y Gauss-Seidel, para evaluar su rendimiento y convergencia.

Tomamos en consideración la ecuación de Poisson siguiente:

$$\begin{cases} -u''(x) = x^2 + 6, & \text{sobre }]0, 1[, \\ u(1) = u(0) = 0. \end{cases} \quad (4.19)$$

Comenzamos por discretizar el intervalo de resolución $]0, 1[$ en $n = 200$ subdivisiones $]x_i, x_{i+1}[$ tal que

$$\forall i \in \{0, \dots, n+1\}, \quad x_i = 0 + i \frac{1}{n+1} = 0 + ih, \quad (4.20)$$

donde n representa las subdivisiones del intervalo $]0, 1[$ en $n+2$ puntos, y h denota la longitud de cada subdivisión. La discretización de la segunda derivada de la ecuación de Poisson se basa en el desarrollo de la serie de Taylor, lo que nos lleva al siguiente problema discreto:

$$\begin{cases} -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = x_i^2 + 6, & i \in \{1, \dots, n\} \\ u_0 = 0, u_{n+1} = 0. \end{cases} \quad (4.21)$$

Donde u_i representa las aproximaciones de $u(x_i)$ y x_i son los puntos discretos en el intervalo $]0, 1[$, y para $i \in \{1, 2, 3, \dots, n\}$, se tiene que $x_i^2 = i^2 h^2$. Ahora, expresamos este sistema de ecuaciones de manera matricial, obteniendo:

$$\underbrace{\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_n \end{bmatrix}}_{\vec{U}} = h^2 \underbrace{\begin{bmatrix} x_1^2 + 6 \\ x_2^2 + 6 \\ x_3^2 + 6 \\ \vdots \\ x_n^2 + 6 \end{bmatrix}}_{\vec{F}}.$$

Esta discretización (4.21) se reduce al sistema $A\vec{U} = \vec{F}$ abordado en (4.8) y el método iterativo SOR se emplea para resolver sistemas de esta forma, en donde los resultados previos (4.9), garantizan la existencia y unicidad de soluciones, proporcionando así las bases esenciales para interpretar los resultados en relación con la discretización y el proceso de resolución. Luego, la función $u(x)$ viene dado por

$$u(x) = -\frac{x^4}{12} - 3x^2 + \frac{37}{12}x$$

A continuación, presentamos gráficas obtenidas a partir de la simulación numérica para resolver el sistema de ecuaciones derivado de la discretización en diferencias finitas para la ecuación de Poisson, en donde se puede observar los resultados obtenidos mediante los métodos iterativos Jacobi, Gauss-Seidel y SOR, con el objetivo de comparar los resultados obtenidos de la solución numérica y la función $u(x)$. Para esto consideramos: $n = 200$, el número de subdivisiones, $\omega = 1.9$ como el parámetro de relajación del método SOR, $tol = 10^{-8}$ como la tolerancia utilizada en la simulación, obteniendo así:

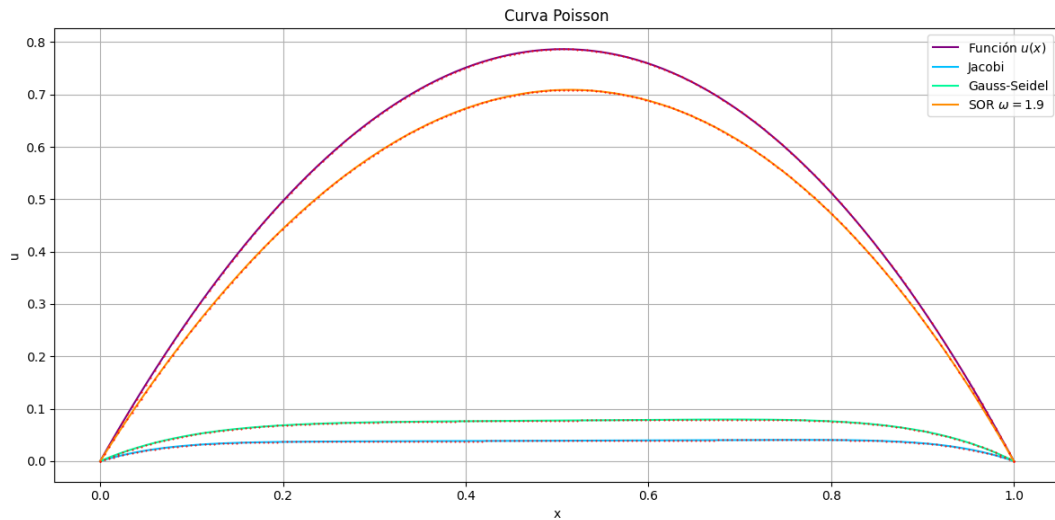


Ilustración 4-1: Análisis de métodos iterativos para la ecuación de Poisson
Realizado por: Yagual, A., 2023.

En la **Ilustración 4-1**, se puede constatar que el método iterativo SOR, con un parámetro de relajación $\omega = 1.9$, exhibe la mejor aproximación a la función $u(x)$, en contraste con el método de Jacobi, que presenta la menor aproximación a la función $u(x)$. Por consiguiente, la elección adecuada del parámetro de relajación mejora significativamente la eficiencia y precisión del método SOR.

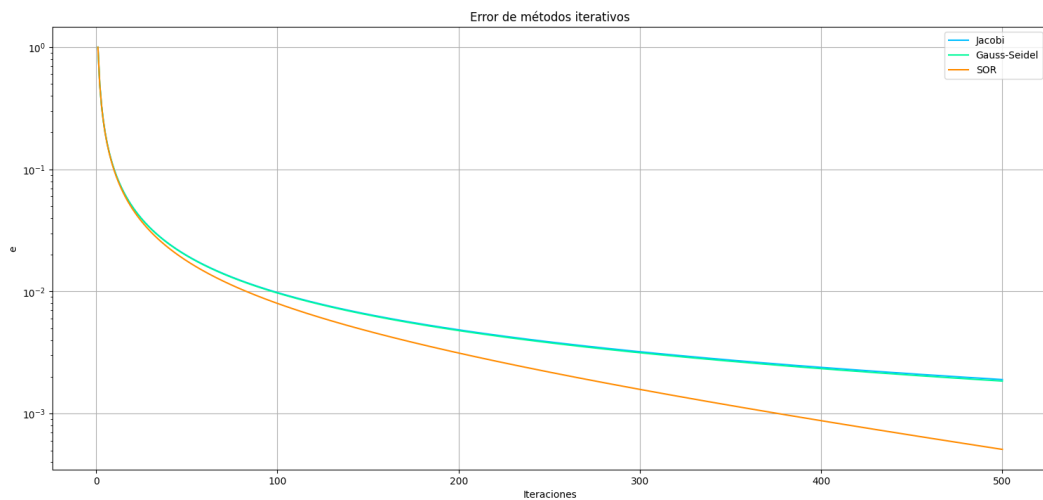


Ilustración 4-2: Error en métodos iterativos para la ecuación de Poisson.
Realizado por: Yagual, A., 2023.

En la **Ilustración 4-2**, se observa que el método iterativo SOR muestra una convergencia más rápida, manifestada por una disminución más pronunciada del error a medida que las iteraciones aumentan, en comparación con los métodos de Jacobi y Gauss-Seidel, los cuales presentan una convergencia más lenta y un error comparativamente mayor.

Tabla 4-1: Errores para cada método iterativo

Tabla de comparación ($n = 200$)			
Métodos Iterativos	Tolerancia	Iteraciones	Error Relativo
Jacobi	10^{-8}	500	0.9508095960712863
Gauss-Seidel	10^{-8}	500	0.9016530878514973
SOR	10^{-8}	500	0.1067702845876724
SOR-Optimizado	10^{-8}	500	0.0030196548952293

Realizado por: Yagual, A., 2023.

La **Tabla 4-1**, revela el rendimiento de distintos métodos iterativos, como son: Jacobi, Gauss-Seidel, SOR y SOR-Optimizado, aplicados en la resolución numérica de la ecuación de Poisson mediante esquemas de diferencias finitas, centrándose en el error relativo, y se observa que a medida que se avanza desde Jacobi hacia SOR, se evidencia una disminución gradual en el error relativo, lo cual pone de manifiesto la mejora en la eficiencia de la convergencia del método SOR en comparación con los otros dos métodos.

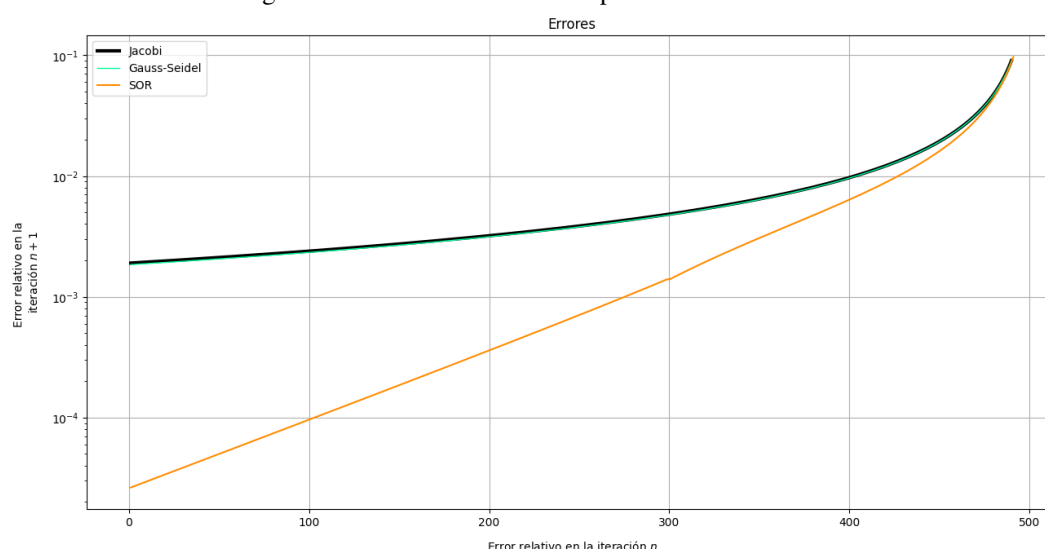


Ilustración 4-3: Análisis del error relativo entre métodos iterativos

Realizado por: Yagual, A., 2023.

En la **Ilustración 3-4**, se evidencia que los métodos de Jacobi y Gauss-Seidel exhiben una pendiente de convergencia cercana a 1, lo que denota una convergencia lineal, dado que el error relativo disminuye aproximadamente a la mitad en cada iteración. Por otro lado, el método SOR presenta una pendiente de convergencia más pronunciada, indicando una convergencia más rápida en comparación con los métodos mencionados previamente.

Tabla 4-2: Análisis del error en métodos iterativos

Tabla de comparación ($n = 200$)				
Métodos Iterativos	Tolerancia	Iteraciones	Error en la Iteración n	Error en la Iteración $n + 1$
Jacobi	10^{-8}	500	0.00480640943462286	0.004782333155395317
Gauss-Seidel	10^{-8}	500	0.00473651928116396	0.004712555702427029
SOR $\omega = 1.9$	10^{-8}	500	0.00139942428833205	0.001379411786240924

Realizado por: Yagual, A., 2023.

En la **Tabla 4-2**, los resultados revelan que el método iterativo SOR exhibe la convergencia más rápida, ya que sus errores son notablemente menores en ambas iteraciones, en contraste con Jacobi y Gauss-Seidel, que muestran una convergencia más lenta, con errores iniciales superiores, señalando una menor eficacia en la aproximación de la solución.

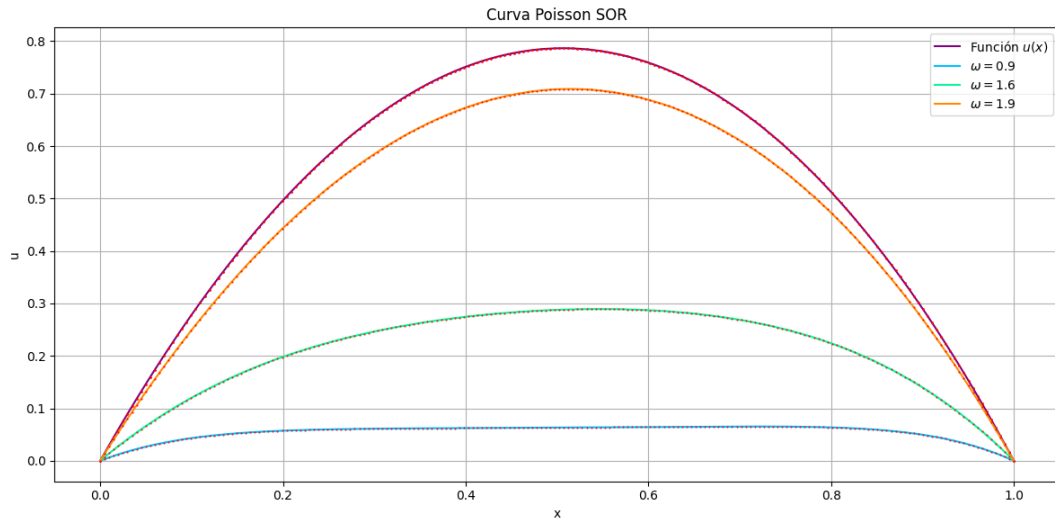


Ilustración 4-4: Análisis de la elección del parámetro ω en la solución SOR
Realizado por: Yagual, A., 2023.

En la **Ilustración 4-4**, se muestra la curva que representa la solución del sistema de ecuaciones lineales provenientes de la discretización de la ecuación de Poisson, utilizando el método iterativo SOR para diferentes valores de ω , en donde se observa que medida que aumenta este valor, la curva se aproxima significativamente hacia la función $u(x)$, exhibiendo un menor error. Esto resalta la importancia de encontrar de ω que maximiza la velocidad y la estabilidad de la convergencia del método.

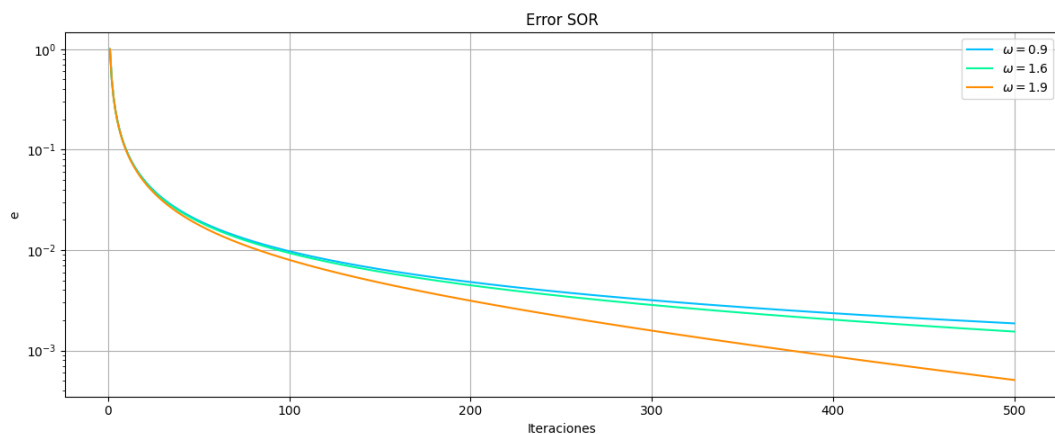


Ilustración 4-5: Error del método iterativo SOR para diferentes valores de ω
Realizado por: Yagual, A., 2023.

La **Ilustración 4-5**, representa el error asociado a cada iteración para los diferentes valores de ω en el método iterativo SOR, en donde se observa que medida que aumenta este valor, el error disminuye significativamente, lo que implica una convergencia más rápida del método.

Sin embargo, por el Teorema 4.1 se sigue que el método SOR converge cuando $0 < \omega < 2$, es importante considerar que valores de ω fuera de este intervalo pueden ocasionar convergencia inestable o divergente, afectando la eficiencia y precisión del método.

Tabla 4-3: Influencia del ω en el error SOR

Tabla de comparación ($n = 200$)			
Método SOR	Tolerancia	Iteraciones	Error Relativo
$\omega = 0.9$	10^{-8}	500	0.9195196136186046
$\omega = 1.6$	10^{-8}	500	0.6358429215586094
$\omega = 1.9$	10^{-8}	500	0.1067702845876724

Realizado por: Yagual, A., 2023.

La **Tabla 4-3** indica la influencia del parámetro de relajación ω en el error del método iterativo SOR, donde se observa que a medida que ω aumenta, el error relativo disminuye considerablemente, lo que resalta la importancia de elegir el valor óptimo de ω para mejorar la velocidad de convergencia en la resolución de sistemas lineales.

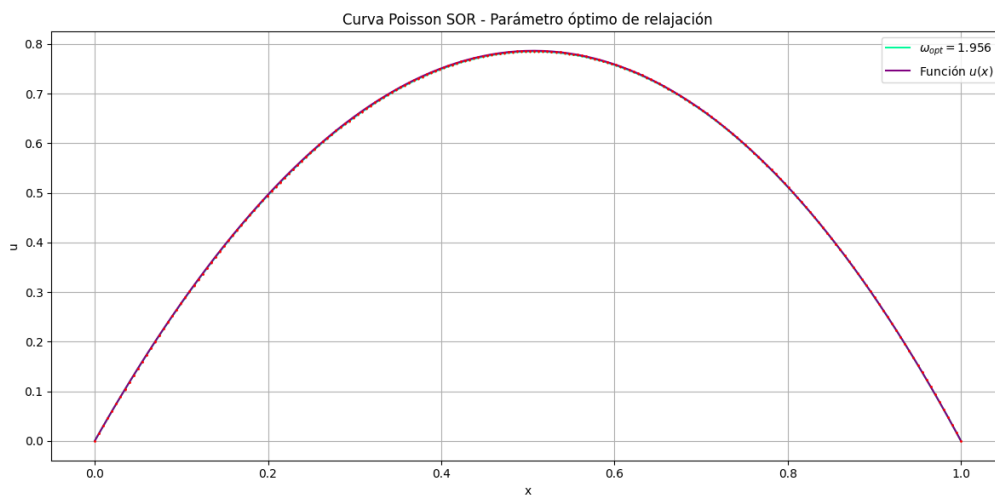


Ilustración 4-6: Análisis del ω_{opt} para el método iterativo SOR

Realizado por: Yagual, A., 2023.

La **Ilustración 4-6** muestra la importancia del parámetro óptimo de relajación ($\omega = 1.9568987275635465$) en el método SOR para la solución al sistema de ecuaciones lineales provenientes de la discretización de la ecuación de Poisson. Se observa que la curva resultante, obtenida mediante el parámetro óptimo, converge de manera precisa hacia la función $u(x)$, lo cual evidencia la capacidad del método iterativo para proporcionar una aproximación precisa y eficiente.

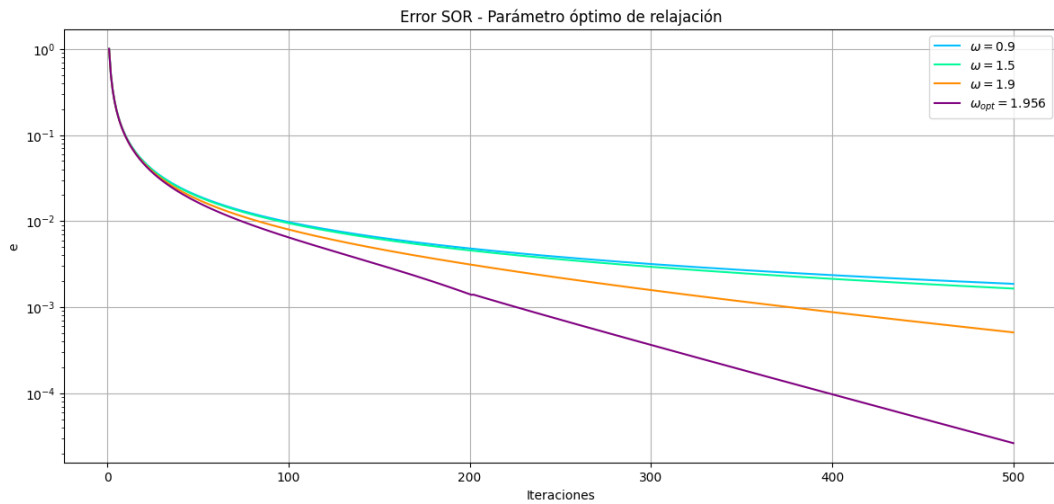


Ilustración 4-7: Error del método iterativo SOR para el ω_{opt}
Realizado por: Yagual, A., 2023.

En la **Ilustración 4-7**, se exhibe la representación gráfica del error para el método iterativo SOR con el parámetro óptimo de relajación ($\omega_{opt} = 1.9568987275635465$), donde se visualiza que a medida que el número de iteraciones aumenta, el error disminuye rápidamente, lo cual indica una rápida convergencia hacia la solución de la función $u(x)$.

Tabla 4-4: Parámetro óptimo del método SOR

n	Valor óptimo de omega	Tolerancia	Error Relativo
25	1.717670306736242	10^{-8}	0.0037636993388401
50	1.842179247258176	10^{-8}	0.0030073807024164
100	1.916337181354349	10^{-8}	0.0030234490329858
200	1.956898727563546	10^{-8}	0.0030196548952293
300	1.970975158141308	10^{-8}	0.0030164854173478
400	1.978121155189865	10^{-8}	0.0030145316611830
500	1.982443707910705	10^{-8}	0.0039088788161940

Realizado por: Yagual, A., 2023.

En la **Tabla 4-4**, se presenta un análisis del comportamiento del parámetro óptimo de relajación del método SOR en función del número de subdivisiones, evidenciando su variación con respecto a diferentes valores, donde se observa un aumento progresivo del valor óptimo de omega a medida que se incrementa este número, siempre respetando la condición de $0 < \omega < 2$ para garantizar la convergencia del método SOR.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

En el apartado que sigue, se presentan las conclusiones y recomendaciones resultantes del estudio sobre la simulación numérica para el parámetro óptimo de relajación del método SOR aplicado a un esquema de diferencias finitas asociado a la ecuación de Poisson, aportando perspectivas valiosas para futuras investigaciones matemáticas.

5.1. CONCLUSIONES

Tras realizar un minucioso análisis detallado y comparativo de los métodos iterativos, como Jacobi, Gauss-Seidel y SOR se concluye que el método SOR se destaca como la opción más adecuada y eficiente, ya que sobresale en términos de velocidad de convergencia y precisión en los resultados obtenidos a partir de la simulación numérica del parámetro óptimo de relajación, además su implementación exitosa en Python ha validado su eficacia en la resolución de sistemas de ecuaciones lineales complejos, demostrando así su relevancia en el análisis numérico y la resolución de problemas en ingeniería y ciencias aplicadas.

El estudio teórico del método iterativo SOR ha permitido comprender la importancia de seleccionar cuidadosamente el valor óptimo del parámetro de relajación en la resolución de sistemas de ecuaciones lineales de gran dimensión, especialmente aquellos derivados de la discretización en diferencias finitas de la ecuación de Poisson, lo que demuestra la capacidad del método SOR para mejorar la eficiencia y la precisión de las soluciones en problemas numéricos complejos.

El análisis detallado de las tablas y gráficos obtenidos de la implementación del método SOR en Python ha permitido observar la variación del valor óptimo del parámetro de relajación en función del radio espectral y la complejidad del sistema de ecuaciones. Además, se ha constatado la necesidad de ajustar el número de iteraciones para garantizar resultados precisos en sistemas más complejos, validando así la utilidad práctica del método SOR en el contexto de la ecuación de Poisson y los esquemas de diferencias finitas asociados.

El análisis exhaustivo de los resultados obtenidos de la implementación del método SOR en Python ha demostrado su eficacia en la resolución de esquemas de diferencias finitas asociados a la ecuación de Poisson. La coherencia y precisión de los resultados validan su utilidad para abordar problemas numéricos desafiantes, resaltando su aplicabilidad en la investigación científica y el desarrollo de soluciones en ingeniería y ciencias aplicadas.

5.2. RECOMENDACIONES

A partir de la búsqueda sistemática y selectiva de referencias pertinentes, se propone ampliar el alcance de la investigación para incluir fuentes de información más actualizadas y diversas, lo que proporcionará una base sólida para la comprensión integral del método SOR y sus aplicaciones en el marco de la ecuación de Poisson y los esquemas de diferencias finitas, además, explorar diferentes fuentes y enfoques permitirá obtener una visión más completa de las diversas perspectivas sobre el parámetro óptimo de relajación y su relevancia en la resolución de problemas numéricos.

Para una mejor percepción de la aplicabilidad del método SOR, se sugiere explorar su implementación en una variedad más amplia de ecuaciones diferenciales ordinarias y parciales, además de la ecuación de Poisson, para evaluar su eficacia en una gama diversa de dominios matemáticos y de ingeniería, lo que contribuirá a una comprensión más profunda de su versatilidad y relevancia en el campo del análisis numérico.

Considerando la implementación exitosa del método SOR en Python para resolver esquemas en diferencias finitas asociados a la ecuación de Poisson, se recomienda explorar técnicas avanzadas de programación y optimización para mejorar aún más la eficiencia y precisión de la implementación, lo que permitirá abordar problemas de mayor complejidad y dimensiones, ampliando así su aplicabilidad en una variedad de ámbitos científicos y de ingeniería.

Para un análisis exhaustivo de los resultados obtenidos, se propone comparar el método SOR con otros enfoques numéricos relevantes, considerando la existencia, unicidad y la convergencia del sistema de ecuaciones, lo que permitirá una evaluación más completa de su eficacia y precisión, identificando las condiciones óptimas para su aplicación y las posibles mejoras en entornos numéricos más desafiantes.

BIBLIOGRAFÍA

BENALCÁZAR, Hernán. *Análisis Numérico*. [en línea]. Quito-Ecuador: Serie de Matemática Universitaria, 2007. [Consulta: 05 junio 2023]. Disponible en: <https://es.scribd.com/document/272668928/Analisis-Numerico-Benalcazar>

DEMMELE, James. *Applied numerical linear algebra* [en línea]. Philadelphia-United State of America: Society for Industrial and Applied Mathematics, 1997. [Consulta: 13 julio 2023]. Disponible en: https://books.google.com.ec/books?id=P3bPAgAAQBAJ&printsec=frontcover&redir_esc=yv=onepage#v=onepage&q&f=false

FAIRES, Douglas. & BURDEN, Richard. *Numerical analysis* [en línea]. 9ª ed. Riad-Arabia Saudita: King Saud University, 2002. [Consulta: 10 agosto 2023]. Disponible en: https://faculty.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf

FRANKEL, Stanley. “Convergence rates of iterative treatments of partial differential equations”. *American Mathematical Society* [en línea], 1950, (United State of America), vol. 4 (30), págs. 65-75. [Consulta: 19 agosto 2023]. ISSN 0025-5718. Disponible en: <https://doi.org/10.1090/S0025-5718-1950-0046149-3>

GREENBAUM, Anne. *Iterative methods for solving linear systems* [en línea]. Philadelphia-United State of America: Society for Industrial and Applied Mathematics, 1997. [Consulta: 04 septiembre 2023]. Disponible en: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611970937.fm>

HACKBUSCH, Wolfgang. *Iterative Solution of Large Sparse Systems of Equations* [en línea]. 2ª ed. New York-United State of America: Springer, 1994. [Consulta: 20 septiembre 2023]. Disponible en: https://www.ciando.com/img/books/extract/3319284835_lp.pdf

QUARTERONI, A. et al. *Cálculo Científico con MATLAB y Octave* [en línea]. Milán-Italia: Springer, 2007. [Consulta: 25 septiembre 2023]. Disponible en: <https://dokumen.tips/documents/calculo-cientifico-con-matlab-y-octave-.html?page=1>

SAAD, Yousef. *Iterative methods for sparse linear systems* [en línea]. Philadelphia-United State of America: Society for Industrial and Applied Mathematics, 2003. [Consulta: 09 octubre 2023]. Disponible en: https://www-users.cse.umn.edu/saad/IterMethBook_2ndEd.pdf

SAMARSKY, Anatolii. *Introducción a los métodos numéricos* [en línea]. Moscú-Rusia: Editorial Mir, 1986. [Consulta: 12 octubre 2023]. Disponible en: https://samarskii.ru/books/book1986_4.pdf

STRANG, Gilbert. *Linear algebra and its applications* [en línea]. 4ª ed. New York-United State of America: Academic Press, 1988. [Consulta: 15 octubre 2023]. Disponible en: https://www.academia.edu/35375838/Linear_algebra_and_its_applications

VARGA, Richard. *Matrix Iterative Analysis* [en línea]. New Jersey-United State of America: Prentice Hall Incorporated, 1962. [Consulta: 16 octubre 2023]. Disponible en: https://www.google.com.ec/books/edition/Matrix_Iterative_Analysis/U2XYs1DyKiYC?hl=en&gbpv=1&pg=PP9&printsec=frontcover

YANG, Shiming. & MATTHIAS, Gobbert. "The optimal relaxation parameter for the SOR method applied to a classical model problem". *Applied Mathematics Letters*, vol. 22, n° 3, (2007), (United State of America). págs. 1-19.

YOUNG, David. "Iterative methods for solving partial difference equations of elliptical type". *American Mathematical Society* [en línea], 1954, (United State of America), vol. 76 (1), págs. 92-111. [Consulta: 20 octubre 2023]. ISSN 0002-9947. Disponible en: <https://doi.org/10.1090/S0002-9947-1954-0059635-7>



ANEXOS

ANEXO A: MÉTODOS ITERATIVOS PARA LA RESOLUCIÓN DE LA ECUACIÓN DE POISSON

```
1 # CARGAR PAQUETES
2 # %matplotlib inline
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from matplotlib.pyplot import figure
6 np.set_printoptions(precision = 4, suppress = True)
7
8 def SOR_Poisson(A, b, x0, tol, iter):
9     n=0; x=x0.copy();
10    D = np.diag(np.diag(A))
11    L = -np.tril(A, -1)
12    U = -np.triu(A, 1)
13    B = np.linalg.inv(D + L).dot(U)
14    w = 2 / (1 + np.sqrt(1 - np.linalg.norm(B, 2)**2))
15    print("Parametro poptimo: relajacion ", w)
16    e=[1]
17    while e[n]>=tol and n<iter:
18        for i in range(len(b)):
19            suma=0;
20            for j in range(len(b)):
21                if i!=j:
22                    suma += A[i,j]*x[j]
23            x[i] = w*(b[i]-suma)/A[i,i]+(1-w)*x[i];
24            e.append(np.linalg.norm(x-x0)/np.linalg.norm(x));
25            n += 1; x0=x.copy();
26    return x, e, n
27
28 def f_jacobi(A,b,x0,tol,iter):
29    e=[1]
30    n=0; x=x0.copy();
31    while e[n]>=tol and n<iter:
32        for i in range(len(b)):
33            suma = 0
34            for j in range(len(b)):
35                if i!=j:
36                    suma += A[i,j]*x0[j]
37            x[i] = (b[i]-suma)/A[i,i];
```

```

38     e.append(np.linalg.norm(x-x0)/np.linalg.norm(x));
39     n += 1; x0=x.copy();
40     return x, e, n
41
42 def f_gauss_seidel(A,b,x0,tol,iter):
43     e=[1]
44     n=0; x=x0.copy();
45     while e[n]>=tol and n<iter:
46         for i in range(len(b)):
47             suma = 0
48             for j in range(len(b)):
49                 if i!=j:
50                     suma += A[i,j]*x[j]
51             x[i] = (b[i]-suma)/A[i,i];
52     e.append(np.linalg.norm(x-x0)/np.linalg.norm(x));
53     n += 1; x0=x.copy();
54     return x, e, n
55
56 def f_SOR(A,b,x0,w,tol,iter):
57     e=[1]
58     n=0; x=x0.copy();
59     while e[n]>=tol and n<iter:
60         for i in range(len(b)):
61             suma=0;
62             for j in range(len(b)):
63                 if i!=j:
64                     suma += A[i,j]*x[j]
65             x[i] = w*(b[i]-suma)/A[i,i]+(1-w)*x[i];
66     e.append(np.linalg.norm(x-x0)/np.linalg.norm(x));
67     n += 1; x0=x.copy();
68     return x, e, n
69 #####   Au = b %%%%%%%%%%
70 # DATOS
71 f = lambda x: x**2+6;
72 ux = lambda x: -x**4/12 - 3*x**2 + 37/12*x
73 ohm = [0, 1];           # Intervalo
74 n = 200 ;               # Nodos internos
75
76 # datos preliminares
77 x = np.linspace(ohm[0], ohm[1], n+2)   # Intervalos completos
78 h = x[1]-x[0]                          # Paso
79 ui = 0; uf = 0;                         # Condiciones de frontera

```

```

80 u0=np.zeros(n); # Condiciones iniciales
81 tol=10**(-8); # Tolerancia
82 iter=500; # Numero max de iteraciones
83 w=1.6; # Omega
84
85 # Llenado de la matriz A y b
86 A=np.zeros([n,n]);
87 b=np.zeros(n);
88 A[0, 0:2] = [2, -1];
89 b[0] = h**2*f(x[1]) - ui;
90 for i in range(1,n-1):
91     A[i, i-1:i+2] = [-1, 2, -1];
92     b[i] = h**2*f(x[i+1]);
93 # ultima fila de A y ultimo elemento de b
94 A[n-1,n-2:n] = [-1, 2];
95 b[n-1] = h**2*f(x[n]) - uf;
96
97 # PROCESO PARA REOSLVER Au=b
98 [u1,e1,n1] = f_jacobi(A,b,u0,tol,iter); met1='Jacobi';
99 [u2,e2,n2] = f_gauss_seidel(A,b,u0,tol,iter); met2='Gauss-Seidel';
100 [u3,e3,n3] = f_SOR(A,b,u0,w,tol,iter); met3='SOR';
101 [u4,e4,n4] = SOR_Poisson(A, b, u0, tol, iter); met4='SOR-poisson';
102 u1 = np.insert(u1, 0, ui)
103 u1 = np.append(u1, uf)
104 e1 = e1[1:]
105 u2 = np.insert(u2, 0, ui)
106 u2 = np.append(u2, uf)
107 e2 = e2[1:]
108 u3 = np.insert(u3, 0, ui)
109 u3 = np.append(u3, uf)
110 e3 = e3[1:]
111
112 # Calcular el error relativo en la iteracion n y n+1
113 enj = e1[:-1] # Error relativo en la iteracion n
114 enj1 = e1[1:] # Error relativo en la iteracion n+1
115 eng = e2[:-1]
116 eng1 = e2[1:]
117 ens = e4[:-1]
118 ens1 = e4[1:]
119
120 # RESULTADO
121 print('--- RESULTADOS -----')

```



```

122 print('-----')
123 print('Solucion del Sistema de Ecuaciones metodo ',met1, 'es: ', u1);
124 print('Con iteraciones ', n1)
125 print('Errores: ', np.array(enj))
126 print('-----')
127 print('Solucion del Sistema de Ecuaciones metodo ',met2, 'es: ', u2);
128 print('Con iteraciones ', n2)
129 print('Errores: ', np.array(eng))
130 print('-----')
131 print('Solucion del Sistema de Ecuaciones metodo ',met3, 'es: ', u3);
132 print('Con iteraciones ', n3)
133 print('-----')
134 print('Solucion del Sistema de Ecuaciones metodo ',met4, 'es: ', u4);
135 print('Con iteraciones ', n4)
136 print('Errores: ', np.array(ens))
137 print("")
138
139 # Grafica de la Funcion
140 pl.plot(x, f(x), label="$f(x)$", color='deepskyblue')
141 pl.plot(x, f(x), 'o', markersize=0, color='red')
142 pl.legend()
143 pl.grid()
144 pl.title('Funcion')
145 pl.show()
146
147 # Grafica de la curva solucion para los metodos iterativos
148 fig, axs = pl.subplots(figsize=(10, 5))
149 colors = ['deepskyblue', 'mediumspringgreen', 'DarkOrange', 'purple']
150 axs.plot(x, ux(x), label="Funcion $u(x)$", color='purple')
151 axs.plot(x, ux(x), 'o', markersize=0.1, color='red')
152 axs.plot(x, u1, label=met1, color=colors[0])
153 axs.plot(x, u1, 'o', markersize=0.1, color='red')
154 axs.plot(x, u2, label=met2, color=colors[1])
155 axs.plot(x, u2, 'o', markersize=0.1, color='red')
156 axs.plot(x, u3, label=met3, color=colors[2])
157 axs.plot(x, u3, 'o', markersize=0.1, color='red')
158 axs.set_ylabel('u')
159 axs.set_xlabel('x')
160 axs.legend()
161 axs.grid()
162 axs.set_title('Curva Poisson')
163 pl.tight_layout()

```

```

164 pl.show()
165
166 # Grafica del error para cada metodo iterativo
167 fig, ax = pl.subplots(figsize=(9, 4))
168 colors = ['deepskyblue', 'mediumspringgreen', 'DarkOrange']
169 ax.semilogy(np.linspace(1, n1, n1), e1, label=met1, color=colors[0])
170 ax.semilogy(np.linspace(1, n1, n1), e1, 'o', markersize=0, color='red')
171 ax.semilogy(np.linspace(1, n2, n2), e2, label=met2, color=colors[1])
172 ax.semilogy(np.linspace(1, n2, n2), e2, 'o', markersize=0, color='red')
173 ax.semilogy(np.linspace(1, n3, n3), e3, label=met3, color=colors[2])
174 ax.semilogy(np.linspace(1, n3, n3), e3, 'o', markersize=0, color='red')
175 ax.set_ylabel('e')
176 ax.set_xlabel('Iteraciones')
177 ax.legend()
178 ax.grid()
179 ax.set_title('Error de metodos iterativos')
180 pl.tight_layout()
181 pl.show()
182
183 # Grafica del error relativo para cada metodo iterativo
184 fig, axs = pl.subplots(figsize=(10,5))
185 xi1 = np.linspace(1, len(e1), len(e1))
186 xi1 = xi1[:len(e1)-10]
187 ei1 = e1[::-1]
188 ei1 = ei1[:len(e1)-10]
189 print(len(xi1))
190 print(len(ei1))
191 axs.semilogy(xi1, ei1, label=met1, color='k', linewidth=3.0)
192 axs.semilogy(xi1, ei1, 'o', markersize=0, color='red')
193
194 # Error Gauss-Seidel
195 xi2 = np.linspace(1, len(e2), len(e2))
196 xi2 = xi2[:len(e2)-10]
197 ei2 = e2[::-1]
198 ei2 = ei2[:len(e2)-10]
199 axs.semilogy(xi2, ei2, label=met2, color='mediumspringgreen', linewidth=1.0)
200 axs.semilogy(xi2, ei2, 'o', markersize=0, color='red')
201
202 # Error SOR
203 xi4 = np.linspace(1, len(e4), len(e4))
204 xi4 = xi4[:len(e4)-10]
205 ei4 = e4[::-1]

```

```

206 ei4 = ei4[:len(e4)-10]
207 axs.semilogy(xi4, ei4, label=met3, color='DarkOrange')
208 axs.semilogy(xi4, ei4, 'o', markersize=0, color='red')
209 axs.set_ylabel('Error relativo en la \niteracion $n+1$', labelpad=10)
210 axs.set_xlabel('Error relativo en la iteracion $n$', labelpad=10)
211 axs.legend()
212 axs.grid()
213 axs.set_title('Errores')
214 pl.subplots_adjust(hspace=0.55)
215 pl.show()
216
217 # DATOS
218 w_values = [0.9, 1.2, 1.9] # PROBAR PARA DIFERENTES VALORES DE OMEGA
219
220 # Grafica de la Curva Poisson SOR para diferentes valores de omega
221 fig1, axs1 = pl.subplots(figsize=(10, 5))
222 print('--- RESULTADOS -----')
223 colors = ['purple', 'deepskyblue', 'mediumspringgreen', 'DarkOrange']
224 line_styles = ['--', '-', '-']
225 marker_styles = ['o', 'o', 'o']
226 # Grafica la funcion u(x) en el primer subplot .
227 axs1.plot(x, ux(x), label="Funcion $u(x)$", color='purple')
228 axs1.plot(x, ux(x), 'o', markersize=1, color='red')
229
230 for i in range(len(w_values)):
231 # PROCESO PARA RESOLVER Au=b
232     [u3, e3, n3] = f_SOR(A, b, u0, w_values[i], tol, iter)
233     met3 = '$\omega=' + str(w_values[i]) + '$'
234     u3 = np.insert(u3, 0, ui)
235     u3 = np.append(u3, uf)
236     e3 = e3[1:]
237
238 # RESULTADO
239     print('Solucion del Sistema de Ecuaciones metodo', met3, 'es:', u3)
240     print('Con iteraciones', n3)
241     print("")
242
243 # Grafica la solucion u3 en el subplot.
244 if w_values[i] == 0.9:
245     axs1.plot(x, u3, label=met3, color='DarkOrange', linestyle=line_styles[i])
246     axs1.plot(x, u3, marker_styles[i], markersize=0, color='DarkOrange')
247 else:

```

```

248     axs1.plot(x, u3, label=met3, color=colors[i], linestyle=line_styles[i])
249     axs1.plot(x, u3, marker_styles[i], markersize=0, color=colors[i])
250
251 # Configuracion del subplot (Curva Poisson SOR).
252 axs1.set_title('Curva Poisson SOR')
253 axs1.set_ylabel('u')
254 axs1.set_xlabel('x')
255 axs1.legend()
256 axs1.grid()
257 pl.tight_layout()
258 pl.show()
259
260 # Grafica del error SOR para diferentes valores de omega
261 fig2, axs2 = pl.subplots(figsize=(9, 4))
262 for i in range(len(w_values)):
263     # PROCESO PARA RESOLVER Au=b
264     [u3, e3, n3] = f_SOR(A, b, u0, w_values[i], tol, iter)
265     met3 = '$\omega=' + str(w_values[i]) + '$'
266     e3 = e3[1:]
267
268 # Grafica el error e3 en escala logarotmica en el subplot.
269 axs2.semilogy(np.linspace(1, n3, n3), e3, label=met3, color=colors[i+1],
270               linestyle=line_styles[i])
271 axs2.semilogy(np.linspace(1, n3, n3), e3, marker_styles[i], markersize=0,
272               color='red')
273
274 # Configuracion del subplot (Error SOR).
275 axs2.set_title('Error SOR')
276 axs2.set_ylabel('e')
277 axs2.set_xlabel('Iteraciones')
278 axs2.legend()
279 axs2.grid()
280 pl.tight_layout()
281 pl.show()
282
283 ##### GRAFICA PARAMETRO OPTIMO DE RELAJACION DEL SOR %%%%%%%%%%
284 # PROCESO PARA REOSLVER Au=b
285 [u4, e4, n4] = SOR_Poisson(A, b, u0, tol, iter)
286 met4 = '$\omega_{opt}=1.956$ '
287 u4 = np.insert(u4, 0, ui)
288 u4 = np.append(u4, uf)
289 e4 = e4[1:]

```

```

290 ens = e4[:-1] # Error relativo en la iteracion n
291 ens1 = e4[1:] # Error relativo en la iteracion n+1
292
293 # RESULTADO
294 print('--- RESULTADOS -----')
295 print('-----')
296 print('Solucion del Sistema de Ecuaciones metodo', met4, 'es:', u4)
297 print('Con iteraciones', n4)
298 print("Con error: ", np.array(ens))
299 print("")
300
301 # GRAFICA 1 - Curva Poisson SOR
302 fig1, axs1 = pl.subplots(figsize=(10, 5))
303 print('--- RESULTADOS -----')
304
305 # Grafica la curva u4 en el primer eje.
306 axs1.plot(x, u4, label=met4, color='deepskyblue')
307 axs1.plot(x, u4, 'o', markersize=0.2, color='red')
308
309 # Grafica la funcion u(x) en el primer eje.
310 axs1.plot(x, ux(x), label="Funcion $u(x)$", color='purple')
311 axs1.plot(x, ux(x), 'o', markersize=0.2, color='red')
312
313 # Configuracion del primer eje.
314 axs1.set_title('Curva Poisson SOR - Parametro optimo de relajacion')
315 axs1.set_ylabel('u')
316 axs1.set_xlabel('x')
317 axs1.legend()
318 axs1.grid()
319 pl.tight_layout()
320 pl.show()
321
322 # Error SOR para el parametro optimo de relajacion
323 fig2, axs2 = pl.subplots(figsize=(10, 5))
324 axs2.semilogy(np.linspace(1, n4, n4), e4, label=met4, color='purple')
325 axs2.semilogy(np.linspace(1, n4, n4), e4, 'o', markersize=0, color='red')
326
327 # Lista de colores para las curvas de error
328 error_colors = ['deepskyblue', 'mediumspringgreen', 'DarkOrange', 'purple']
329 for i in range(len(w_values)):
330 # PROCESO PARA RESOLVER Au=b
331 [u3, e3, n3] = f_SOR(A, b, u0, w_values[i], tol, iter)

```

```
332 met3 = '$\omega=' + str(w_values[i]) + '$'
333 e3 = e3[1:]
334
335 # Grafica el error e3 en escala logaritmica en el subplot.
336 axs2.semilogy(np.linspace(1, n3, n3), e3, label=met3, color=error_colors[i],
337 linestyle=line_styles[i])
338 axs2.semilogy(np.linspace(1, n3, n3), e3, marker_styles[i], markersize=0,
339 color=error_colors[i])
340
341 # Configuracion del segundo eje.
342 axs2.set_title('Error SOR - Parametro optimo de relajacion')
343 axs2.set_ylabel('e')
344 axs2.set_xlabel('Iteraciones')
345 axs2.legend()
346 axs2.grid()
347 pl.tight_layout()
348 pl.show()
```



epoch

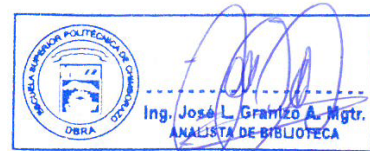
**Dirección de Bibliotecas y
Recursos del Aprendizaje**

**UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y
DOCUMENTAL**

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 28 / 11 / 2023

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: ALEX YASSER YAGUAL GONZÁLEZ
INFORMACIÓN INSTITUCIONAL
Facultad: CIENCIAS
Carrera: MATEMÁTICA
Título a optar: MATEMÁTICO
f. Analista de Biblioteca responsable: ING. JOSÉ LIZANDRO GRANIZO ARCOS MGRT.



1998-DBRA-UPT-2023