UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO
DIVISIÓN ACADÉMICA DE CIENCIAS BÁSICAS

RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE POISSON EN 1D Y 2D POR EL MÉTODO DE DIFERENCIAS FINITAS

TESIS

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN MATEMÁTICAS

PRESENTA

EDWIN ENRIQUE PÉREZ RODRIGUEZ

DIRECTOR

DR. JUSTINO ALAVEZ RAMÍREZ

Cunduacán, Tab. Abril 2021

DIRECCIÓN

Cunduacán, Tabasco; a 23 de marzo de 2021.

C. EDWIN ENRIQUE PÉREZ RODRÍGUEZ
PASANTE DE LA LIC. EN MATEMÁTICAS
P R E S E N T E

Por medio del presente y de la manera más cordial, me dirijo a usted para hacer de su conocimiento que proceda a la impresión del trabajo titulado **"RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE POISSON EN 1D Y 2D POR EL MÉTODO DE DIFERENCIAS FINITAS"** bajo la modalidad de titulación por **TESIS**, en virtud de que reúne los requisitos para el **EXAMEN PROFESIONAL** correspondiente.

Sin otro particular, reciba usted un cordial saludo.

A T E N T A M E N T E

DR. GERARDO DELGADILLO PIÑÓN
DIRECTOR

C.c.p. Pasante.
C.c.p. Archivo.

Km.1 Carretera Cunduacán-Jalpa de Méndez, A.P. 24, C.P. 86690, Cunduacán, Tab., México.
Tel/Fax: (993) 3581500 Ext. 6702,6701 E-Mail: direccion.dacb@ujat.mx
www.ujat.mx

CARTA DE AUTORIZACIÓN

El que suscribe, autoriza por medio del presente escrito a la Universidad Juárez Autónoma de Tabasco para que utilice tanto física como digitalmente la tesis de grado denominada "**RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE POISSON EN 1D Y 2D POR EL MÉTODO DE DIFERENCIAS FINITAS**", de la cual soy autor y titular de los Derechos de Autor.

La finalidad del uso por parte de la Universidad Juárez Autónoma de Tabasco de la tesis antes mencionada, será única y exclusivamente para difusión, educación y sin fines de lucro; autorización que se hace de manera enunciativa mas no limitativa para subirla a la Red Abierta de Bibliotecas Digitales (RABID) y a cualquier otra red académica con las que la Universidad tenga relación institucional.

Por lo antes manifestado, libero a la Universidad Juárez Autónoma de Tabasco de cualquier reclamación legal que pudiera ejercer respecto al uso y manipulación de la tesis mencionada y para los fines estipulados en este documento.

Se firma la presente autorización en la ciudad de Villahermosa, Tabasco a los 22 días del mes de Abril del año 2021.

AUTORIZÓ

EDWIN ENRIQUE PÉREZ RODRÍGUEZ
152A11013

Índice general

Dedicatoria y Agradecimientos

Quiero dedicar esta tesis a mi papá, Victor Manuel Pérez García; en cada momento de mi vida como estudiante me animaba con su amor y cariño. Al salir de la preparatoria me miró a los ojos para decirme que le echáramos muchas ganas para lograr que me graduara de la universidad: él con su esfuerzo y yo con mi dedicación. Hoy puedo decirle con mucho orgullo: "¡Lo logramos Papá!" Aunque no me puede acompañar físicamente en estos momentos, celebra conmigo desde algún lugar del espacio divino.

A mi mamá, Concepción, por haber confiado en mí y apoyarme en cada una de mis decisiones; su amor, cariño y ejemplo de perseverancia me animó a continuar en esos momentos en los cuales quería desistir.

A mis hermanos, Edgar y Sofía, por animarme y acompañarme en mis días de estrés.

Agradecimientos:

A todos mis amigos y compañeros que me ayudaron de una manera desinteresada, muchas gracias por toda su ayuda y buena voluntad.

Quiero agradecer especialmente al Dr. Justino por haber confiado y creído en mí, por su paciencia y generosidad al compartir su conocimiento y motivarme a seguir formándome como científico y matemático.

También quiero agradecer a la Universidad Juárez Autónoma de Tabasco, a la División Académica de Ciencias Básicas y a mis profesores, quienes con su servicio, dedicación y paciencia contribuyeron a mi formación como profesionista y matemático.

Finalmente quiero agradecer a Dios por haber puesto en mí la semilla de la curiosidad, el espíritu crítico y la generosidad de compartir lo aprendido.

Prólogo

En el presente trabajo se busca encontrar la solución de la ecuación de Poisson

$$-\Delta u = f(x), \quad x \in \operatorname{int}\Omega$$

donde,

$$\Delta = \frac{\partial^2}{\partial x_1^2},\ \Omega \subset \mathbb{R}$$

o

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2},\ \Omega \subset \mathbb{R}^2,$$

mediante un método numérico. Esta ecuación es frecuentemente encontrada en áreas de la ciencia como física, química, biología, etc., al tratar de explicar el comportamiento de procesos estacionarios de distinta naturaleza física, es decir, fenómenos que son independientes del tiempo, como oscilaciones, conducción de calor, difusión, entre muchos otros [7]. La importancia de aprender a resolver la ecuación de Poisson mediante un método numérico se hace presente cuando no es posible encontrar una solución analítica en un punto de interés.

Se ha escogido el "Método de Diferencias Finitas", el cual consiste en discretizar el dominio de la ecuación de Poisson mediante un mallado, en un número finito de puntos; de esta manera, el problema de resolver una ecuación diferencial parcial se convierte en encontrar la solución de un sistema de ecuaciones lineales algebraicas de la forma $Ax = b$.

Para hallar la solución de $Ax = b$ se estudian los métodos iterativos clásicos y su convergencia: Jacobi, Gauss-Seidel y de Relajación. La matriz A obtenida en la ecuación de Poisson en una dimensión tiene la forma de una matriz tridiagonal, por lo que también se estudia un método directo propio para este tipo de matrices: el método de Thomas.

El capítulo 1 de la tesis trata sobre los métodos iterativos en general. Se han utilizado como referencia los libros de *Numerical Analysis: A Second Course* de Ortega (1972)[5], *Analysis of Numerical Methods* de Isaacson y Keller (1966)[4] e *Introduction to Numerical Analysis* de Stoer y Bulirsch (1993)[9].

En el capítulo 2 se analizan y comparan los métodos iterativos de Jacobi, Gauss-Seidel y de Relajación; además, se implementan los algoritmos computacionales de cada uno de ellos en el lenguaje de programación de OCTAVE. Los libros que se han utilizado como referencia son: *Análisis Numérico* de Burden y Faires (1998)[1], *Matrix Iterative Analysis* de Vargas (1962)[11], *Numerical Mathematics* de Quarteroni, Sacco y Saleri (2007)[6], *Analysis of Numerical Methods* de Isaacson y Keller (1966)[4], *Introduction to Numerical Analysis* de Stoer y Bulirsch (1993)[9], *Iterative Solution of Large Linear Systems* de Young (1971)[12] y *Métodos y Esquemas Numéricos: Un análisis computacional* de Skiva (2005)[8].

El capítulo 3 se estudia el método de Thomas y se compara con los métodos iterativos vistos en el capítulo 2; también se ha implementado el algoritmo computacional de este método en el lenguaje de programación de OCTAVE. Se han utilizado como referencia los libros de *Numerical Mathematics* de Quarteroni, Sacco y Saleri (2007)[6] y *An Introduction to Numerical Analysis* de Süli y Mayers (2003)[10].

En el capítulo 4, sección 4.1. se estudia de manera breve la ecuación de Poisson; posteriormente, en la sección 4.2. y 4.3. se analiza como aplicar el método de "Diferencias Finitas" en la ecuación de Poisson en una y dos dimensiones. Finalmente, en la sección 4.4. se resuelve un ejemplo numérico aplicando el método de "Diferencias Finitas". Los libros que se han utilizado como referencia son: *Applied Numerical Lineal Algebra* de Demmel (1997)[2], *Ecuaciones de la Física Matemática* de Tijonov y Samarsky (1980)[7], *Introduction to Numerical Analysis* de Stoer y Bulirsch (1993)[9], *Iterative Solution of Large Linear Systems* de Young (1971)[12] y *The eigenproblem of a tridiagonal 2-Toeplitz matrix* de Gover (1994)[3].

40
En el capítulo 5 se tienen las conclusiones y posibles trabajos futuros.

Capítulo 1

Introducción a los métodos iterativos

En este capítulo estudiaremos qué son los métodos iterativos usados para aproximar la solución de un sistema lineal de ecuaciones algebraicas dado por

$$Ax = b, \tag{1.1}$$

donde $A \in \mathbb{R}^{n\times n}$ es no singular y $b \in \mathbb{R}^n$. Aquí $\mathbb{R}^{m\times n}$ denota el espacio de las matrices $m \times n$ con entradas reales. Analizaremos qué condiciones se requieren para su convergencia; revisaremos también cómo se construyen los métodos iterativos y su estabilidad. Por último, observaremos cómo medir el error de aproximación y calcular una cota para el mismo. Las referencias principales son: *Numerical Analysis: A Second Course* de Ortega (1972)[5], *Analysis of Numerical Methods* de Isaacson y Keller (1966)[4] e *Introduction to Numerical Analysis* de Stoer y Bulirsch (1993)[9].

1.1. Motivación

Un método iterativo para resolver el sistema (1.1) consiste en construir una sucesión $\{x^{(k)}\}_{k=0}^{\infty}$ de vectores en $\mathbb{R}^n$ que aproximan a la solución exacta $x = A^{-1}b$. Así que lo que se quiere es encontrar dicha solución con la propiedad

$$\lim_{k\to\infty} x^{(k)} = x\,.$$

Los métodos iterativos construyen la sucesión iterativa citada de la siguiente forma: se elige una matriz $B \in \mathbb{R}^{n\times n}$, un vector $c \in \mathbb{R}^n$, un punto de inicio $x^{(0)} \in \mathbb{R}^n$, y se itera

$$x^{(k+1)} = Bx^{(k)} + c\,, \quad k \geq 0\,. \tag{1.2}$$

A B le llamaremos **matriz de iteración** del método iterativo (1.2).

Definición 1.1.1. *El método iterativo* (1.2) *se dice* ***convergente*** *si para cualquier* $x^{(0)} \in \mathbb{R}^n$, *se verifica la igualdad*

$$\lim_{k\to\infty} x^{(k)} = x = A^{-1}b\,. \tag{1.3}$$

Si el método (1.2) es convergente, entonces se debe tener

$$x = \lim_{k\to\infty} x^{(k)} = \lim_{k\to\infty} \left(Bx^{(k-1)} + c\right) = B \lim_{k\to\infty} x^{(k-1)} + c = Bx + c$$

lo que implica

$$c = (I - B)x = (I - B)A^{-1}b\,.$$

Definición 1.1.2. *El método iterativo* (1.2) *se dice* ***consistente*** *si* $c = (I - B)A^{-1}b$.

El concepto de radio espectral de una matriz que definiremos a continuación, nos será de mucha utilidad para probar la convergencia de los métodos iterativos.

Definición 1.1.3. *Se llama* ***radio espectral*** *de* $B \in \mathbb{R}^{n\times n}$ *a*

$$\rho(B) = \max_{\lambda \text{ valor propio de } B} |\lambda|\,.$$

Ejemplo 1.1.1. *Sea* $B = \begin{bmatrix} 4 & 2 \\ 6 & 3 \end{bmatrix}$. *Calcula* $\rho(B)$.
Primero calculamos el polinomio característico de B:

$$\begin{aligned} p(\lambda) &= \left| \begin{bmatrix} 4 & 2 \\ 6 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| \\ &= \begin{vmatrix} 4-\lambda & 2 \\ 6 & 3-\lambda \end{vmatrix} \\ &= (4-\lambda)(3-\lambda) - 12 \\ &= 12 - 7\lambda + \lambda^2 - 12 \\ &= -7\lambda + \lambda^2, \end{aligned}$$

cuyas raíces son $\lambda_1 = 0$ *y* $\lambda_2 = 7$, *por lo que* $\rho(B) = \max\{0, 7\} = 7$.

Para demostrar la convergencia del método (1.2) presentamos algunos resultados del álgebra matricial que nos serán de utilidad.

Definición 1.1.4. *Una* $B \in \mathbb{R}^{n\times n}$ *es* ***convergente*** *si*

$$\lim_{k\to\infty} B^k = 0.$$

Lema 1.1.1. *Sea $B \in \mathbb{R}^{n\times n}$, entonces B es convergente si y sólo si*

$$\lim_{k\to\infty} B^k y = 0,$$

para cualquier $y \in \mathbb{R}^n$.

Demostración. *a*) Bajo el supuesto de que la norma matricial es consistente con la norma vectorial, es decir, $\|B^k y\| \leq \|B^k\| \cdot \|y\|$ $\forall y \in \mathbb{R}^n$ y que $\lim_{k\to\infty} B^k = 0$, se sigue que

$$\lim_{k\to\infty} \|B^k y\| \leq \lim_{k\to\infty} \|B^k\| \cdot \|y\| = \|y\| \lim_{k\to\infty} \|B^k\| = \|y\| \cdot \| \lim_{k\to\infty} B^k\| = \|y\| \cdot 0 = 0.$$

Por lo tanto, $\lim_{k\to\infty} \|B^k y\| = 0 \; \forall \; y \in \mathbb{R}^n$, es decir, $\| \lim_{k\to\infty} B^k y\| = 0 \; \forall \; y \in \mathbb{R}^n$, lo que implica que $\lim_{k\to\infty} B^k y = 0 \; \forall \; y \in \mathbb{R}^n$.
b) Supongamos ahora que $\lim_{k\to\infty} B^k y = 0$ para cualquier $y \in \mathbb{R}^n$. Sea $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ el $i-$ésimo vector unitario en $\mathbb{R}^n$. Puesto que

$$B^k = B^k I = B^k[e_1, e_2, \ldots, e_n] = [B^k e_1, B^k e_2, \ldots, B^k e_n], \quad k \geq 1,$$

y $\lim_{k\to\infty} B^k e_i = 0$ para todo $i = 1 : n$, se sigue que existe el $\lim_{k\to\infty} B^k$ y vale 0. Así que B es convergente. □

Lema 1.1.2 (Ortega (1972)[5])**.** *Sea $B \in \mathbb{R}^{n\times n}$ y $\varepsilon > 0$. Entonces, existe una norma matricial consistente y submultiplicativa*[1] *$\|\cdot\|$ que depende de B y de ε, tal que*

$$\rho(B) < \|B\| \leq \rho(B) + \varepsilon.$$

Demostración. Sea $J = P^{-1}BP$, donde J es la forma de Jordan de B y $P \in \mathbb{R}^{n\times n}$ no singular. Sean

$$D = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \varepsilon & 0 & \cdots & 0 \\ 0 & 0 & \varepsilon^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \varepsilon^{n-1} \end{bmatrix} \quad \text{y} \quad \widehat{J} = D^{-1}JD = \begin{bmatrix} \widehat{J}_1 & 0 & \cdots & 0 \\ 0 & \widehat{J}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widehat{J}_m \end{bmatrix}$$

donde cada $\widehat{J}_i = \lambda_i$ si λ_i es un valor propio de multiplicidad uno de B, y

$$\widehat{J}_i = \begin{bmatrix} \lambda_i & \varepsilon & \cdots & 0 \\ 0 & \lambda_i & \ddots & \vdots \\ \vdots & \vdots & \ddots & \varepsilon \\ 0 & 0 & \cdots & \lambda_i \end{bmatrix}$$

[1] $\|AB\| \leq \|A\|\|B\|$ para toda $A, B \in \mathbb{R}^{n\times n}$.

si λ_i es un valor propio de multiplicidad k de B, $1 < k < n$.
Ahora, por un lado tenemos que

$$\|\widehat{J}\|_\infty = \max_{1\le r\le n} \sum_{s=1}^{n} |j_{rs}| = |\lambda_i| \le \rho(B) + \varepsilon,$$

o bien

$$\|\widehat{J}\|_\infty = \max_{1\le r\le n} \sum_{s=1}^{n} |j_{rs}| = |\lambda_i| + |\varepsilon| \le \rho(B) + \varepsilon.$$

Sea $Q = PD$ y consideremos $\|x\| = \|Q^{-1}x\|_\infty \ \forall x \in \mathbb{R}^n$ que es una norma en $\mathbb{R}^n$ (véase proposición 1.2.3 de Ortega (1972)[5]). Entonces

$$\begin{aligned} \|B\| &= \max_{\|x\|=1} \|Bx\| = \max_{\|Q^{-1}x\|_\infty=1} \|Q^{-1}Bx\|_\infty = \max_{\|y\|_\infty=1} \|Q^{-1}BQy\|_\infty \\ &= \max_{\|y\|_\infty=1} \|\widehat{J}y\|_\infty = \|\widehat{J}\|_\infty \le \rho(B) + \varepsilon. \end{aligned}$$

□

Un resultado de gran utilidad práctica es el siguiente:

Teorema 1.1.1 (Isaacson and Keller (1996)[4]). *Si $B \in \mathbb{R}^{n\times n}$, entonces*

$$\rho(B) \le \|B\|$$

para cualquier norma matricial consistente y submultiplicativa $\|\cdot\|$ en $\mathbb{R}^{n\times n}$. De hecho se cumple la igualdad

$$\rho(B) = \inf_{\|\cdot\|} \|B\|.$$

Demostración. Sea λ un valor propio de B. Entonces existe un vector propio $x \neq 0$ tal que

$$Bx = \lambda x.$$

Así

$$\|\lambda x\| = \|Bx\| \le \|B\|\|x\|.$$

Como $\|\lambda x\| = |\lambda|\|x\|$, entonces $|\lambda| \le \|B\|$.
Por lo tanto,

$$\rho(B) = \max_{\lambda \text{ valor propio de } B} |\lambda| \le \|B\|.$$

Del lema 1.1.2 se concluye que

$$\rho(B) = \inf_{\|\cdot\|} \|B\|.$$

□

Lema 1.1.3 (Isaacson and Keller (1966)[4], Quarteroni *et al.* (2007)[6]). *Sea $B \in \mathbb{R}^{n\times n}$, entonces B es convergente si y sólo si $\rho(B) < 1$.*

Demostración. a) Si $\rho(B) < 1$, entonces existe $\varepsilon > 0$ tal que $\rho(B) < 1 - \varepsilon$, por lo tanto $\rho(B)+\varepsilon < 1$, y por el lema 1.1.2 existe una norma matricial $\|\cdot\|$ consistente y submultiplicativa tal que

$$\rho(B) < \|B\| \leq \rho(B) + \varepsilon.$$

De donde se sigue que $\|B\| < 1$. Entonces $\lim_{k\to\infty} \|B\|^k = 0$. Como $0 \leq \|B^k\| \leq \|B\|^k$, entonces $\lim_{k\to\infty} \|B^k\| = 0$, de donde a su vez se sigue $\lim_{k\to\infty} B^k = 0$, por lo que B es convergente.

b) Supongamos que B es convergente. Sea λ un valor propio de B y $x \neq 0$ el vector propio asociado a λ, es decir, $Bx = \lambda x$. Se tiene que

$$\begin{aligned} B^k x &= B^{k-1}(Bx) \\ &= B^{k-1}(\lambda x) \\ &= \lambda^2 (B^{k-2} x) \\ &\vdots \\ &= \lambda^k x. \end{aligned}$$

De la hipótesis y por el lema 1.1.1, se sigue que

$$\lim_{k\to\infty} B^k x = 0.$$

Por lo que

$$\lim_{k\to\infty} \lambda^k x = 0,$$

de donde

$$\lim_{k\to\infty} \lambda^k = 0.$$

Por lo tanto $|\lambda| < 1$ y, debido a que λ es un valor propio cualquiera de B, se sigue que $\rho(B) < 1$.

□

A continuación probaremos la convergencia del método (1.2).

Teorema 1.1.2 (De convergencia: Stoer and Bulirsch (1993)[9]). *El método* (1.2) *es convergente si y sólo si es consistente y* $\rho(B) < 1$.

Demostración. a) Supongamos que el método (1.2) es convergente. Se sigue de la definicion 1.1.2 que es consistente. Hay que demostrar que $\rho(B) < 1$. Definimos el error en la $k-$ésima iteración como

$$e^{(k)} = x^{(k)} - x, \quad k \geq 0. \tag{1.4}$$

De (1.2) se tiene que $x^{(k)} = Bx^{(k-1)} + c$, y por hipótesis se sigue que $x = Bx + c$, entonces (1.4) se puede escribir como

$$\begin{aligned} e^{(k)} &= Bx^{(k-1)} + c - Bx - c \\ &= B(x^{(k-1)} - x) \\ &= Be^{(k-1)} \\ &= B^2e^{(k-2)} \\ &\vdots \\ &= B^ke^{(0)}, \quad k \geq 1. \end{aligned}$$

Por lo tanto,

$$e^{(k)} = B^ke^{(0)}. \tag{1.5}$$

Por hipótesis

$$\lim_{k\to\infty} e^{(k)} = 0,$$

por lo que

$$\lim_{k\to\infty} B^ke^{(0)} = 0.$$

Del lema 1.1.1 se sigue que B es convergente, y finalmente el lema 1.1.3 nos dice que $\rho(B) < 1$.

b) Supongamos ahora que el método (1.2) es consistente y $\rho(B) < 1$. Procedemos por contraposición, suponiendo que $\rho(B) \geq 1$.
Si $\rho(B) > 1$, por definición de $\rho(B)$ debe existir algún valor propio λ tal que $|\lambda| > 1$. Sea $e^{(0)}$ un vector propio de B asociado a λ, es decir, $Be^{(0)} = \lambda e^{(0)}$.
Si elegimos a $e^{(0)}$ como el error inicial de la iteración, es decir,

$$\begin{aligned} e^{(0)} &= x^{(0)} - x \\ e^{(1)} &= x^{(1)} - x \\ &\vdots \\ e^{(k)} &= x^{(k)} - x \\ &\vdots \end{aligned}$$

Por (1.5) se tiene que

$$e^{(k)} = B^k e^{(0)} = \lambda^k e^{(0)}.$$

Luego

$$\lim_{k\to\infty} \|e^{(k)}\| = \|e^{(0)}\| \lim_{k\to\infty} |\lambda|^k = \infty, \text{ ya que } |\lambda| > 1.$$

Por lo que el método (1.2) no es convergente.

Ahora, si $\rho(B) = 1$, por definición de $\rho(B)$ debe existir algún valor propio λ tal que $|\lambda| = 1$. Sea $e^{(0)}$ un vector propio de B asociado a λ, es decir, $Be^{(0)} = \lambda e^{(0)}$. De nuevo por (1.5) se sigue que

$$\begin{aligned} e^{(k)} &= B^k e^{(0)} = \lambda^k e^{(0)} \\ \|e^{(k)}\| &= |\lambda|^k \|e^{(0)}\| = \|e^{(0)}\| \\ \lim_{k\to\infty} \|e^{(k)}\| &= \|e^{(0)}\|. \end{aligned}$$

Por lo que $\lim_{k\to\infty} \|e^{(k)}\|$ existe pero se mantiene constante de valor $\|e^{(0)}\|$, lo cual nos dice que la iteración $x^{(k)}$ se mantiene constante de valor $x^{(0)}$, por lo que no converge a x. Por lo tanto, el método (1.2) no es convergente. □

1.2. Construcción de los métodos iterativos

Una alternativa para elegir la matriz de iteración B y construir el método iterativo (1.2) es tomar una descomposición regular de la matriz A como

$$A = M - N \tag{1.6}$$

con M no singular. Así que el sistema (1.1) se puede escribir en la forma

$$\begin{aligned} (M - N)x &= b \\ Mx &= Nx + b \\ x &= M^{-1}Nx + M^{-1}b, \end{aligned}$$

lo que sugiere estudiar el método iterativo

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b, \quad k \geq 0. \tag{1.7}$$

Comparando (1.7) con (1.2), se sigue que la matriz de iteración del método (1.7) es

$$B = M^{-1}N$$

y

$$c = M^{-1}b = M^{-1}AA^{-1}b = M^{-1}(M-N)A^{-1}b = (I - M^{-1}N)A^{-1}b = (I-B)A^{-1}b\,,$$

por lo que el método (1.7) es consistente. Por el Teorema 1.1.2, para que el método sea convergente, es suficiente que $\rho(M^{-1}N) < 1$.
Aplicando el Teorema 1.1.1, se concluye que para que un método del tipo (1.7) sea convergente es suficiente que sea consistente y que $\|M^{-1}N\| < 1$, para alguna norma matricial consistente y submultiplicativa $\|\cdot\|$ en $\mathbb{R}^{n\times n}$. Además, se tiene este otro resultado no menos importante:

Teorema 1.2.1. *Todo método iterativo convergente del tipo* (1.2) *es de la forma* (1.7) *con* $A = M - N$*, siendo* M *no singular.*

Demostración. Como el método (1.2) se puede tomar de la forma (1.7) se tiene que $B = M^{-1}N$ y $N = MB$. Ahora construyamos M tomando la descomposición regular de la matriz A:

$$\begin{aligned} A &= M - N \\ &= M - MB \\ &= M(I-B). \end{aligned}$$

Para que el sistema (1.1) tenga solución, A tiene que ser no singular y como M es no singular, se sigue que $I - B$ también es no singular. Por lo tanto,

$$M = A(I-B)^{-1}.$$

□

1.3. Análisis de estabilidad y error de aproximación

Estabilidad. Para analizar la estabilidad del método iterativo (1.2), se supondrá que se están realizando los cálculos cometiendo un pequeño error $\varepsilon^{(k)}$ en cada paso. Así que por un lado, los cálculos exactos serían

$$x^{(k+1)} = Bx^{(k)} + c\,, \quad k \geq 0$$

y por otro lado, los cálculos aproximados serían

$$\tilde{x}^{(k+1)} = B\tilde{x}^{(k)} + c + \varepsilon^{(k)}\,, \quad k \geq 0\,.$$

Así que el error que se cometerá en cada iteración será

$$\begin{aligned} \tilde{x}^{(k+1)} - x^{(k+1)} &= B\left(\tilde{x}^{(k)} - x^{(k)}\right) + \varepsilon^{(k)}\,, \quad k \geq 0 \\ \tilde{x}^{(0)} - x^{(0)} &= \varepsilon^{(0)}\,. \end{aligned}$$

Notemos que

$$\begin{aligned}\tilde{x}^{(k+1)} - x^{(k+1)} &= B(\tilde{x}^{(k)} - x^{(k)}) + \varepsilon^{(k)} \\ &= B[B(\tilde{x}^{(k-1)} - x^{(k-1)}) + \varepsilon^{(k-1)}] + \varepsilon^{(k)} \\ &= B^2(\tilde{x}^{(k-1)} - x^{(k-1)}) + B\varepsilon^{(k-1)} + \varepsilon^{(k)} \\ &\vdots \\ &= B^{k+1}\varepsilon^{(0)} + B^k\varepsilon^{(0)} + B^{k-1}\varepsilon^{(1)} + \cdots + B\varepsilon^{(k-1)} + \varepsilon^{(k)}\,,\end{aligned}$$

y bajo el supuesto de que se está trabajando con una norma matricial submultiplicativa en $\mathbb{R}^{n\times n}$ y subordinada a una norma vectorial en $\mathbb{R}^n$ (se utilizará la misma notación $\|\cdot\|$ para ambas normas), y si además $\|B\| < 1$, se sigue que

$$\begin{aligned}\|\tilde{x}^{(k+1)} - x^{(k+1)}\| &\leq \|B^{k+1}\varepsilon^{(0)}\| + \|B^k\varepsilon^{(0)}\| + \|B^{k-1}\varepsilon^{(1)}\| + \cdots + \|B\varepsilon^{(k-1)}\| + \|\varepsilon^{(k)}\| \\ &\leq \|B^{k+1}\|\,\|\varepsilon^{(0)}\| + \|B^k\|\,\|\varepsilon^{(0)}\| + \|B^{k-1}\|\,\|\varepsilon^{(1)}\| + \cdots + \\ &\quad \|B\|\,\|\varepsilon^{(k-1)}\| + \|\varepsilon^{(k)}\| \\ &\leq \max_{0\leq j\leq k}\|\varepsilon^{(j)}\| \sum_{j=0}^{k+1}\|B\|^j \\ &\leq \max_{0\leq j\leq k}\|\varepsilon^{(j)}\| \sum_{j=0}^{\infty}\|B\|^j \\ &= \max_{0\leq j\leq k}\|\varepsilon^{(j)}\| \frac{1}{1-\|B\|}.\end{aligned}$$

Si $\max\limits_{0\leq j\leq k}\|\varepsilon^{(j)}\| < u$, u la unidad de redondeo de la aritmética de punto flotante que se está trabajando, entonces

$$\|\tilde{x}^{(k+1)} - x^{(k+1)}\| \leq \frac{u}{1-\|B\|}\,,$$

por lo que el método es estable siempre que sea convergente ya que pequeños errores de redondeo producen errores pequeños en la iteración $k+1$, que además se pueden hacer bastante pequeños si $\|B\| \ll 1$.

1.4. Cota para el error de aproximación

Bajo el supuesto de que $\lim\limits_{k\to\infty} x^{(k)} = x$ y por consiguiente $c = (I-B)A^{-1}b = (I-B)x$, se sigue para $k \geq 1$:

$$\|x^k - x\| = \|Bx^{k-1} + c - x\| = \|Bx^{k-1} + (I-B)x - x\| = \|Bx^{k-1} - Bx\| \leq \|B\|\|x^{k-1} - x\|\,,$$

por lo que

$$\begin{aligned} \|x^{(k)} - x\| &\leq \|B\|\|x^{(k-1)} - x\| \\ &= \|B\| \, \|x^{(k-1)} - x^{(k)} + x^{(k)} - x\| \\ &\leq \|B\| \, \|x^{(k-1)} - x^{(k)}\| + \|B\| \, \|x^{(k)} - x\| \, , \end{aligned}$$

de donde

$$[1 - \|B\|]\|x^{(k)} - x\| \leq \|B\| \, \|x^{(k)} - x^{(k-1)}\|$$

Si $0 < \|B\| < 1$, entonces

$$\|x^{(k)} - x\| \leq \frac{\|B\|}{1 - \|B\|}\|x^{(k)} - x^{(k-1)}\| \, , \quad k \geq 1 \, , \tag{1.8}$$

por lo que si se quiere aproximar x con una tolerancia $TOL > 0$, es suficiente tomar

$$\varepsilon = \frac{1 - \|B\|}{\|B\|} TOL$$

e iterar hasta que $\|x^{(k)} - x^{(k-1)}\| < \varepsilon$, y esto implicará que $\|x^{(k)} - x\| < TOL$.

Capítulo 2

Algunos métodos iterativos clásicos

En este capítulo estudiaremos los métodos iterativos clásicos de Jacobi, Gauss-Seidel y Relajación; analizaremos cómo construirlos y bajo qué condiciones convergen. Implementamos el algoritmo computacional de cada uno de ellos en el lenguaje de programación Octave; así como una comparación entre ellos para alcanzar convergencia. Las principales referencias son: *Análisis Numérico* de Burden y Faires (1998)[1], *Matrix Iterative Analysis* de Vargas (1962)[11], *Numerical Mathematics* de Quarteroni, Sacco y Saleri (2007)[6], *Analysis of Numerical Methods* de Isaacson y Keller (1966)[4], *Introduction to Numerical Analysis* de Stoer y Bulirsch (1993)[9], *Iterative Solution of Large Linear Systems* de Young (1971)[12] y *Métodos y Esquemas Numéricos: Un análisis computacional* de Skiva (2005)[8].

Como se dijo en la sección 1.2 del Capítulo 1, los metodos iterativos convergentes que estamos considerando en este trabajo para resolver un sistema lineal $Ax = b$ es de la forma

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b\,, \quad k \geq 0.$$

Equivalentemente

$$Mx^{(k+1)} = Nx^{(k)} + b\,, \quad k \geq 0\,,$$

donde $A = M - N$ con M no singular.
Una alternativa para elegir las matrices M y N cuando A no tiene ceros en su diagonal principal es considerar la descomposición

$$A = D - L - U\,, \tag{2.1}$$

donde

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & 0 & \cdots & 0 \\ -a_{31} & -a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & -a_{n3} & \cdots & 0 \end{bmatrix}$$

y

$$U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1,n-1} & -a_{1n} \\ 0 & 0 & \cdots & -a_{2,n-1} & -a_{2n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Algunos métodos iterativos clásicos eligen las matrices M y N a partir de las matrices D, L y U, convenientemente, como veremos en las secciones que siguen.

2.1. Método de Jacobi

El método de Jacobi, bajo la descomposición (1.6) y (2.1), consiste en elegir

$$M = D \qquad \text{y} \qquad N = L + U\,.$$

Así que este método se expresa como

$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b\,, \quad k \geq 0\,, \tag{2.2}$$

donde la matriz de iteración (del método de Jacobi), que la denotaremos como E_J, es

$$E_J = D^{-1}(L+U). \tag{2.3}$$

Ahora, usando componentes, el método toma la forma:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1,\, j\neq i}^{n} a_{ij}x_j^{(k)}\right), \quad k \geq 0\,, \tag{2.4}$$

bajo el supuesto que $a_{ii} \neq 0$, $i = 1:n$.

Ejemplo 2.1.1. *Calcular las primeras tres aproximaciones de la solución de $Ax = b$ usando el método de Jacobi, donde*

$$A = \begin{bmatrix} -3/10 & 1/10 & 0 & 0 \\ 1/10 & 1/10 & -1/10 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 2 \\ 0 \\ 3 \end{bmatrix}$$

y el punto de inicio

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Solución. Primero factoricemos la matriz A como $A = D - L - U$, donde

$$D = \begin{bmatrix} -3/10 & 0 & 0 & 0 \\ 0 & 1/10 & 0 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1/10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & -1/10 & 0 & 0 \\ 0 & 0 & 1/10 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Para aplicar (2.2), calculamos

$$D^{-1}(L+U) = \begin{bmatrix} -10/3 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10/7 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 0 & -1/10 & 0 & 0 \\ -1/10 & 0 & 1/10 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1/3 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

y

$$D^{-1}b = \begin{bmatrix} -1/3 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10/7 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 6 \\ 2 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} -20 \\ 20 \\ 0 \\ 15 \end{bmatrix}.$$

Así que las iteraciones serán

$$x^{(1)} = D^{-1}(L+U)x^{(0)} + D^{-1}b = \begin{bmatrix} -20 \\ 20 \\ 0 \\ 15 \end{bmatrix},$$

$$x^{(2)} = \begin{bmatrix} 0 & 1/3 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -20 \\ 20 \\ 0 \\ 15 \end{bmatrix} + \begin{bmatrix} -20 \\ 20 \\ 0 \\ 15 \end{bmatrix} = \begin{bmatrix} 20/3 \\ 20 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 20 \\ 20 \\ 0 \\ 15 \end{bmatrix} = \begin{bmatrix} -40/3 \\ 40 \\ 0 \\ 15 \end{bmatrix},$$

$$x^{(3)} = \begin{bmatrix} 0 & 1/3 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -40/3 \\ 40 \\ 0 \\ 15 \end{bmatrix} + \begin{bmatrix} -20 \\ 20 \\ 0 \\ 15 \end{bmatrix} = \begin{bmatrix} 40/3 \\ 40/3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -20 \\ 20 \\ 0 \\ 15 \end{bmatrix} = \begin{bmatrix} -20/3 \\ 100/3 \\ 0 \\ 15 \end{bmatrix}.$$

2.2. Implementación computacional del método de Jacobi

El criterio de paro que se va a implementar es

$$\frac{\|x^{(k+1)} - x^{(k)}\|_\infty}{\|x^{(k+1)}\|_\infty + eps} < tol, \tag{2.5}$$

donde *eps* es el epsilon de la maquina y *tol* es la tolerancia dada por el usuario. La siguiente función está escrita en *OCTAVE*.

```
function Jacobi(A,b,x0,tol,itermax)
%
% Esta función implementa el Método de Jacobi para aproximar la solución
%   % de Ax = b de acuerdo con (1.7). El criterio de paro es
%    Error = norm(x_max-x, inf)/(norm(x_max, inf) + eps) < tol
% cuando hay convergencia.
%
%%%%%%%%%%%%%%%%%%%% Datos de entrada %%%%%%%%%%%%%%%%%%%%
% A es una matriz cuadrada no singular de tamaño n x n.
% b es un vector de tamaño n x 1.
% x0 es el punto de inicio.
% itermax es el número máximo de iteraciones permitidas.
%
%%%%%%%%%%%%%%%%%%%% Datos de salida %%%%%%%%%%%%%%%%%%%%
% k es el número de iteraciones.
% x_max es el vector en la iteración k
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Universidad Juárez Autónoma de Tabasco.
% Fecha:19/septiembre/2018.
% Autores: Edwin E.,Justino A., Victoria O.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
x0 = x0(:); b = b(:);
B = -1 * A + diag(diag(A));  % donde B = L + U de acuerdo con (1.7).
D = diag(diag(A).^-1) ;    % donde D es la inversa de D en (1.7).
P = D * B;
Q = D * b;
% Método de Jacobi.
x = x0;
 for k = 1:itermax
```

```
  k, x_max = P * x + Q
  Error = norm(x_max - x, inf)/(norm(x_max, inf) + eps);
  if Error < tol
   break
  end
  x = x_max;
 endfor
endfunction
```

Ejemplo 2.2.1. *Aproximar la solución de $Ax = b$ del ejemplo 2.1.1 con una tolerancia de 10^{-7} usando el método de Jacobi.*

Solución. Aplicamos la función **Jacobi.m** en *OCTAVE*, cuya sintaxis es

$$\textbf{Jacobi}(A, b, x^{(0)}, tol, itermax)$$

donde $x^{(0)}$ es el punto de inicio del método, *tol* es la tolerancia para el criterio de paro e *itermax* es el número máximo de iteracciones permitidas en caso de que no hubiera convergencia.
En la tabla 2.1 presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un *itermax* de 50.

$x^{(0)}$	k	$x^{(k)}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999303082806 \\ 30.000000696917191 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999326313377 \\ 30.000000696917191 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999303082806 \\ 30.000000766608913 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$

Tabla 2.1: Aproximaciones de la solución de $Ax = b$ del ejemplo 2.2.1, donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada.

Ejemplo 2.2.2. *Aproximar la solución de* $Ax = b$ *donde*

$$A = \begin{bmatrix} -0.3 & 0.1 & 0.03 & 0.1 \\ 0.1 & 0.1 & 0 & 0.1 \\ 0 & -0.2 & 0.7 & 0 \\ 0.2 & 0.01 & 0 & 0.2 \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} 6 \\ 2 \\ 0 \\ 3 \end{bmatrix},$$

con una tolerancia de 10^{-5} *usando el método de Jacobi.*

Solución. En la tabla 2.2 presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un *itermax* de 200.

$x^{(0)}$	k	$x^{(k)}$
$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$	104	$\begin{bmatrix} -9.88719881721810 \\ 5.26333100520705 \\ 1.50379130188842 \\ 24.62419684090217 \end{bmatrix}$
$\begin{bmatrix} 2 \\ -3 \\ -5 \\ 0 \end{bmatrix}$	104	$\begin{bmatrix} -9.88721181012231 \\ 5.26337494083686 \\ 1.50936671762453 \\ 24.62421778179110 \end{bmatrix}$
$\begin{bmatrix} 0 \\ -4 \\ 1 \\ 3 \end{bmatrix}$	104	$\begin{bmatrix} -9.88720821975179 \\ 5.26334353365531 \\ 1.50378679165595 \\ 24.62419842864593 \end{bmatrix}$

Tabla 2.2: Aproximaciones de la solución de $Ax = b$ del ejemplo 2.2.2, donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada.

2.3. Método de Gauss-Seidel

Este método consiste, de acuerdo con la descomposición (1.6) y (2.1), en tomar $M = D - L$ y $N = U$, por lo que su matriz de iteración, que denotaremos por E_{GS}, será

$$E_{GS} = (D - L)^{-1}U, \tag{2.6}$$

y se expresa como una de las tres opciones siguientes:

$$\begin{aligned} x^{(k+1)} &= (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b\,, \quad k \geq 0. \\ (D-L)x^{(k+1)} &= Ux^{(k)} + b\,, \quad k \geq 0. \end{aligned} \tag{2.7}$$

O en términos de componentes, bajo el supuesto que $a_{ii} \neq 0$, $i = 1:n$,

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}\left(b_1 - \sum_{2}^{n} a_{1j}x_j^{(k)}\right), \quad k \geq 0\,, \\ x_i^{(k+1)} &= \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 2:n-1, \quad k \geq 0\,, \\ x_n^{(k+1)} &= \frac{1}{a_{nn}}\left(b_n - \sum_{j=1}^{n-1} a_{nj}x_j^{(k+1)}\right), \quad k \geq 0\,, \end{aligned}\right\} \tag{2.8}$$

Ejemplo 2.3.1. *Calcular las primeras tres aproximaciones de la solución de $Ax = b$ usando el método Gauss-Seidel, donde A y b son los dados en el ejemplo 2.1.1 con el punto de inicio*

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Solución. Primero factoricemos la matriz A como $A = D - L - U$, donde D, L y U son como las dadas en el ejemplo 2.1.1. Para aplicar (2.7), calculamos

$$D-L = \begin{bmatrix} -3/10 & 0 & 0 & 0 \\ 0 & 1/10 & 0 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1/10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -3/10 & 0 & 0 & 0 \\ 1/10 & 1/10 & 0 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix}.$$

Entonces

$$(D-L)^{-1} = \begin{bmatrix} -10/3 & 0 & 0 & 0 \\ 10/3 & 10 & 0 & 0 \\ 0 & 0 & 10/7 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}.$$

$$(D-L)^{-1}b = \begin{bmatrix} -10/3 & 0 & 0 & 0 \\ 10/3 & 10 & 0 & 0 \\ 0 & 0 & 10/7 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 6 \\ 2 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} -20 \\ 40 \\ 0 \\ 15 \end{bmatrix},$$

y

$$(D-L)^{-1}U=\begin{bmatrix}-10/3 & 0 & 0 & 0\\ 10/3 & 10 & 0 & 0\\ 0 & 0 & 10/7 & 0\\ 0 & 0 & 0 & 5\end{bmatrix}\begin{bmatrix}0 & -1/10 & 0 & 0\\ 0 & 0 & 1/10 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}=\begin{bmatrix}0 & 1/3 & 0 & 0\\ 0 & -1/3 & 1 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}.$$

Así que las primeras tres iteraciones serán:

$$x^{(1)}=(D-L)^{-1}Ux^{(0)}+(D-L)^{-1}b=\begin{bmatrix}0 & 1/3 & 0 & 0\\ 0 & -1/3 & 1 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\0\\0\\0\end{bmatrix}+\begin{bmatrix}-20\\40\\0\\15\end{bmatrix}=\begin{bmatrix}-20\\40\\0\\15\end{bmatrix},$$

$$x^{(2)}=\begin{bmatrix}0 & 1/3 & 0 & 0\\ 0 & -1/3 & 1 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}-20\\40\\0\\15\end{bmatrix}+\begin{bmatrix}-20\\40\\0\\15\end{bmatrix}=\begin{bmatrix}-40/3\\-40/3\\0\\0\end{bmatrix}+\begin{bmatrix}-20\\40\\0\\15\end{bmatrix}=\begin{bmatrix}-20/3\\80/3\\0\\15\end{bmatrix},$$

y

$$x^{(3)}=\begin{bmatrix}0 & 1/3 & 0 & 0\\ 0 & -1/3 & 1 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}-20/3\\80/3\\0\\15\end{bmatrix}+\begin{bmatrix}-20\\40\\0\\15\end{bmatrix}=\begin{bmatrix}80/9\\-80/9\\0\\0\end{bmatrix}\begin{bmatrix}-20\\40\\0\\15\end{bmatrix}=\begin{bmatrix}-100/9\\280/9\\0\\15\end{bmatrix}.$$

2.4. Implementación computacional del método de Gauss-Seidel

El criterio de paro que se va a implementar es el mismo que se utilizó en la implementación del método de Jacobi que está definido en (2.5). La siguiente función está escrita en *OCTAVE*.

```
function GaussSeidel(A,b,x0,tol,itermax)
%
%%%%%%%%%%%%%%% Datos de entrada %%%%%%%%%%%%%%%%
```

```
% A es una matriz cuadrada de tamaño n x n.
% b es un vector de tamaño n x 1.
% x0 es el punto de inicio
% itermax es el número de interacciones máximas.
%
%%%%%%%%%%%%%%%% Datos de salida %%%%%%%%%%%%%%%%%%
% k el número de iteración.
% x_max es el vector en la iteración k
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Universidad Juárez Autónoma de Tabasco.
% Fecha:19/septiembre/2018.
% Autores: Edwin E.,Justino A., Victoria O.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Descomposición de A = D - L - U,
clc
format long
x0 = x0(:); b = b(:);
D = diag(diag(A));
L = -tril(A, -1);
U = -triu(A, 1);
B = D - L;
%% Método de Gauss-Seidel.
x = x0;
for k = 1:itermax
k, p = U * x + b;
x_max = linsolve(B,p)
Error = norm(x_max-x, inf)/(norm(x_max, inf) + eps);
if Error < tol
break
end
x = x_max;
endfor
endfunction
```

Ejemplo 2.4.1. *Aproximar la solución de* $Ax = b$ *del ejemplo 2.1.1 con una tolerancia de* 10^{-7} *usando el método de Gauss-Seidel, y comparar los resultados con los obtenidos en el método de Jacobi en el ejemplo 2.2.1.*

Solución. Aplicamos la función **GaussSeidel.m** en *OCTAVE*, cuya sintaxis es

$$\textbf{GaussSeidel}(A, b, x^{(0)}, tol, itermax)$$

donde $x^{(0)}$ es el punto de inicio del método, *tol* es la tolerancia para el criterio de paro e *itermax* es el número máximo de iteraciones permitidas en caso de que no hubiera convergencia.

En la tabla 2.3 presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un *itermax* de 50 y los comparamos con los resultados obtenidos con el método de Jacobi. Se observa que el método de Gauss-Seidel solo requiere para su convergencia de aproximadamente la mitad de iteraciones que requiere el método de Jacobi.

	Método de Gauss-Seidel		Método de Jacobi	
$x^{(0)}$	k	$x^{(k)}$	k	$x^{(k)}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	16	$\begin{bmatrix} -9.999999303082808 \\ 29.999999303082809 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999303082806 \\ 30.000000696917191 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	16	$\begin{bmatrix} -9.999999319344209 \\ 29.999999319344212 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999326313377 \\ 30.000000696917191 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$	16	$\begin{bmatrix} -9.999999333282553 \\ 29.999999333282553 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999303082806 \\ 30.000000766608913 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$

Tabla 2.3: Comparación de los métodos iterativos de Gauss-Seidel y Jacobi en la solución de $Ax = b$ del ejemplo 2.4.1, donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada. Notamos que el método de Gauss-Seidel converge más rápido que el método de Jacobi.

Ejemplo 2.4.2. *Aproximar la solución de $Ax = b$ donde A y b están dados en el ejemplo 2.2.2 con una tolerancia de 10^{-5} usando el método de Gauss-Seidel, y comparar con los obtenidos en el ejemplo 2.2.2.*

Solución. En la tabla 2.4 presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un *itermax* de 200 y se comparan con los obtenidos con el método de Jacobi. Se observa en este caso que el método de Gauss-Seidel no converge desde ninguno

	Método de Gauss-Seidel		Método de Jacobi	
$x^{(0)}$	k	$x^{(k)}$	k	$x^{(k)}$
$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$	200	$\begin{bmatrix} -6.450042031159100 \\ -1.304655228319636 \\ -0.372758636662753 \\ 21.515274792575081 \end{bmatrix}$	104	$\begin{bmatrix} -9.88719881721810 \\ 5.26333100520705 \\ 1.50379130188842 \\ 24.62419684090217 \end{bmatrix}$
$\begin{bmatrix} 2 \\ -3 \\ -5 \\ 0 \end{bmatrix}$	200	$\begin{bmatrix} -6.119605004510659 \\ -1.936059797406291 \\ -0.553159942116083 \\ 21.216407994380972 \end{bmatrix}$	104	$\begin{bmatrix} -9.88721181012231 \\ 5.26337494083686 \\ 1.50936671762453 \\ 24.62421778179110 \end{bmatrix}$
$\begin{bmatrix} 0 \\ -4 \\ 1 \\ 3 \end{bmatrix}$	200	$\begin{bmatrix} -6.550379663603693 \\ -1.112928410597560 \\ -0.317979545885017 \\ 21.606026084133568 \end{bmatrix}$	104	$\begin{bmatrix} -9.88720821975179 \\ 5.26334353365531 \\ 1.50378679165595 \\ 24.62419842864593 \end{bmatrix}$

Tabla 2.4: Comparación del método de Gauss-Seidel con el método de Jacobi en la solución de $Ax = b$ del ejemplo 2.4.2, donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada. Notamos que en este caso, el método de Jacobi converge a la tolerancia 10^{-5} en 104 iteraciones, mientras que el método de Gauss-Seidel no logra convergencia desde ninguno de los tres puntos de inicio.

de los tres puntos de inicio dados, mientras que el método de Jacobi converge a la tolerancia deseada. En la sección 2.5 explicaremos porqué ocurre esto.

Ejemplo 2.4.3. *Aproximar la solución de $Ax = b$, donde*

$$A = \begin{bmatrix} 0.3 & 0.2 & 0.05 & 0.03 \\ 0 & 0.4 & 0.21 & 0.1 \\ 0.01 & 0.02 & -0.2 & 0.01 \\ 0.05 & 0.03 & 0.01 & 0.1 \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} 4 \\ -3 \\ 2 \\ -1 \end{bmatrix},$$

con una tolerancia de 10^{-7} y usando los métodos de Jacobi y Gauss-Seidel.

Solución. Aplicamos las funciones **Jacobi.m** y **GaussSeidel.m** en *OCTAVE*. En la tabla 2.5 presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un *itermax* de 50. Notamos que el método de *Gauss-Seidel* converge en un número menor de iteraciones a la solución de $Ax = b$, en contraste con el número de iteraciones que realiza del método de *Jacobi*. De hecho se observa que el método de Gauss-Seidel converge aproximadamente 1.3 veces más rápido que el método de Jacobi.

	Método de Gauss-Seidel		Método de Jacobi	
$x^{(0)}$	k	$x^{(k)}$	k	$x^{(k)}$
$\begin{bmatrix} 4 \\ 7 \\ 3 \\ -9 \end{bmatrix}$	23	$\begin{bmatrix} 14.03753316951147 \\ 5.80837980725438 \\ -17.23612011506060 \\ -17.03766851542599 \end{bmatrix}$	31	$\begin{bmatrix} 14.03753313860342 \\ 5.80837965150036 \\ -17.23612001363965 \\ -17.03766891617553 \end{bmatrix}$
$\begin{bmatrix} 2 \\ -4 \\ -3 \\ 1 \end{bmatrix}$	22	$\begin{bmatrix} 14.03753367985832 \\ 5.80837982958711 \\ -17.23611979186144 \\ -17.03766880961915 \end{bmatrix}$	28	$\begin{bmatrix} 14.03753318056049 \\ 5.80837961176520 \\ -17.23611961545371 \\ -17.03766894216591 \end{bmatrix}$
$\begin{bmatrix} -6 \\ 2 \\ 0 \\ -5 \end{bmatrix}$	22	$\begin{bmatrix} 14.03753415577187 \\ 5.80838014879701 \\ -17.23611931172817 \\ -17.03766919135222 \end{bmatrix}$	30	$\begin{bmatrix} 14.03753317766840 \\ 5.80838000233856 \\ -17.23611915242265 \\ -17.03766932960663 \end{bmatrix}$

Tabla 2.5: Comparación del método de Gauss-Seidel con el método de Jacobi en la solución de $Ax = b$ del ejemplo 2.4.1, donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada. En este caso, observamos que el método de Gauss-Seidel converge más rápido que el método de Jacobi.

2.5. Convergencia de los métodos de Jacobi y Gauss-Seidel

Observamos en el ejemplo 2.4.2 que el método de Gauss-Seidel no converge mientras que el método de Jacobi si lo hace, y en los ejemplos 2.4.1 y 2.4.3 ambos métodos convergen pero Gauss-Seidel converge más rápido que el método de Jacobi. Veremos condiciones que garantizan la convergencia de los métodos.

Definición 2.5.1. *Se dice que* $A \in \mathbb{R}^{n\times n}$ *es* ***estrictamente diagonal dominante*** *si*

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|$$

para todo $i = 1 : n$.

El siguiente teorema, cuya demostración se puede consultar en las páginas 68–70 del libro de Vargas (1962) [11], será utilizado para demostrar la convergencia de los métodos, ya que se puede observar, por uno de los incisos, que cuando un método converge, entonces ambos lo hacen, y el método de Gauss-Seidel converge más rápido que el de Jacobi.

Teorema 2.5.1 (Stein-Rosenberg, Vargas (1962) [11]). *Si $E_J = D^{-1}(L+U)$ es la matriz de iteración del método de Jacobi y $E_{GS} = (D-L)^{-1}U$ es la matriz de iteración del método de Gauss-Seidel, entonces será válida una y solo una de las siguientes afirmaciones:*

a) $\rho(E_J) = \rho(E_{GS}) = 0$.

b) $0 \leq \rho(E_{GS}) < \rho(E_J) < 1$.

c) $\rho(E_J) = \rho(E_{GS}) = 1$.

d) $1 < \rho(E_J) < \rho(E_{GS})$.

Ejemplo 2.5.1. *Analizar los radios espectrales de las matrices de iteración de los ejemplos 2.1.1, 2.2.2 y 2.4.3, y explicar la conducta de la convergencia en cada caso.*

Solución.
a) La matriz de coeficientes del sistema lineal del ejemplo 2.1.1 es

$$A = \begin{bmatrix} -3/10 & 1/10 & 0 & 0 \\ 1/10 & 1/10 & -1/10 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix}.$$

Por lo que las matrices de iteración de los métodos de Jacobi y Gauss-Seidel son

$$E_J = D^{-1}(L+U) = \begin{bmatrix} 0 & 1/3 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{y} \quad E_{GS} = (D-L)^{-1}U = \begin{bmatrix} 0 & 1/3 & 0 & 0 \\ 0 & -1/3 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

de donde

$$\rho(E_J) = 0.5774 \quad \text{y} \quad \rho(E_{GS}) = 0.3333.$$

Como $\rho(E_J) > \rho(E_{GS})$, se cumple la parte b) del teorema 2.5.1, por lo que el método de Gauss-Seidel converge más rápido que el método Jacobi. Esto se observa en los resultados que se muestran en la tabla 2.3.
b) La matriz de coeficientes del sistema lineal del ejemplo 2.2.2 es

$$A = \begin{bmatrix} -0.3 & 0.1 & 0.03 & 0.1 \\ 0.1 & 0.1 & 0 & 0.1 \\ 0 & -0.2 & 0.7 & 0 \\ 0.2 & 0.01 & 0 & 0.2 \end{bmatrix}.$$

De manera que las matrices de iteración de los métodos de Jacobi y Gauss-Seidel son

$$E_J = D^{-1}(L+U) = \begin{bmatrix} 0 & 1/3 & 1/10 & 1/3 \\ -1 & 0 & 0 & -1 \\ 0 & 2/7 & 0 & 0 \\ -1 & -1/20 & 0 & 0 \end{bmatrix}$$

y

$$E_{GS} = (D-L)^{-1}U = \begin{bmatrix} 0 & 1/3 & 1/10 & 1/3 \\ 0 & -1/3 & -1/10 & -4/3 \\ 0 & -2/21 & -1/35 & -8/21 \\ 0 & -19/60 & -19/200 & -4/15 \end{bmatrix},$$

de donde

$$\rho(E_J) = 0.8945 \quad \text{y} \quad \rho(E_{GS}) = 0.9930.$$

Por ser ambos menores que 1, el teorema 1.1.2 nos garantiza que los dos métodos siguen siendo convergentes, sin embargo, el método de Gauss-Seidel converge más lentamente como se observa en la tabla 2.4.

c) Ahora la matriz de coeficientes del sistema lineal del ejemplo 2.4.3 es

$$A = \begin{bmatrix} 0.3 & 0.2 & 0.05 & 0.03 \\ 0 & 0.4 & 0.21 & 0.1 \\ 0.01 & 0.02 & -0.2 & 0.01 \\ 0.05 & 0.03 & 0.01 & 0.1 \end{bmatrix}.$$

En este caso, las matrices de iteración de los métodos de Jacobi y Gauss-Seidel son

$$E_J = D^{-1}(L+U) = \begin{bmatrix} 0 & -2/3 & -1/6 & -1/10 \\ 0 & 0 & -21/40 & -1/4 \\ 1/20 & 1/10 & 0 & 1/20 \\ -1/2 & -3/10 & -1/10 & 0 \end{bmatrix}$$

y

$$E_{GS} = (D-L)^{-1}U = \begin{bmatrix} 0 & -2/3 & -1/6 & -1/10 \\ 0 & 0 & -21/40 & -1/4 \\ 0 & -1/30 & -73/1200 & 1/50 \\ 0 & 101/300 & 2963/12000 & 123/1000 \end{bmatrix},$$

de donde

$$\rho(E_J) = 0.3950 \quad \text{y} \quad \rho(E_{GS}) = 0.2740.$$

Como $\rho(E_J) > \rho(E_{GS})$, se cumple la parte b) del teorema 2.5.1. Por esta razón el método de Gauss-Seidel converge a la solución más rápido que el método de Jacobi, lo cual se puede observar en la tabla 2.5. Notemos que la matriz A es estrictamente diagonal dominante, por lo que el teorema 2.5.2 siguiente también garantiza que ambos métodos son convergentes.

Teorema 2.5.2 (Carteroni *et al.* (2007) [6]). *Si $A \in \mathbb{R}^{n\times n}$ es estrictamente diagonal dominante, entonces los métodos de Jacobi y Gauss-Seidel son convergentes. Se cumple que $\|E_{GS}\| \leq \|E_J\| < 1$.*

Demostración. Recordemos que el teorema 1.1.2 nos garantiza que el método iterativo (1.2) es convergente si y sólo si es consistente y $\rho(E_J) < 1$.
Para el método de Jacobi (2.2):

$$E_J = D^{-1}(L+U) \quad \text{y} \quad c = D^{-1}b.$$

Puesto que

$$E_J = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix}.$$

Entonces por ser A estrictamente diagonal dominante, se sigue que

$$\|E_J\|_\infty = \max_{1\leq i\leq n} \sum_{j\neq i} \left|\frac{a_{ij}}{a_{ii}}\right| = \max_{1\leq i\leq n} \frac{1}{|a_{ii}|} \sum_{j\neq i} |a_{ij}| < \frac{|a_{ii}|}{|a_{ii}|} = 1.$$

Como el teorema 1.1.1 nos dice que $\rho(E_J) \leq \|E_J\|_\infty$, por lo tanto, $\rho(E_J) < 1$. Así el método de Jacobi es convergente.
Para el método de Gauss-Seidel (2.7): $E_{GS} = (D-L)^{-1}U$ y $c = (D-L)^{-1}b$. Como $\rho(E_J) < 1$, se sigue del teorema 2.5.1(b) que $\rho(E_{GS}) < \rho(E_J) < 1$. De modo que $\rho(E_{GS}) < 1$. Por lo tanto, el método de Gauss-Seidel también es convergente. □

Teorema 2.5.3 (Isaacson and Keller (1996)[4]). *Si $A, M \in \mathbb{R}^{n\times n}$, donde A es simétrica y M no singular, para el cual*

$$Q \equiv M + M^T - A \tag{2.9}$$

es definida positiva, entonces

$$B \equiv I - M^{-1}A$$

es convergente si y sólo si A es definida positiva.

Demostración. Supongamos que A es definida positiva. Sea λ un valor propio de B y $u \in \mathbb{R}^n$ un vector propio asociado a λ, así que $Bu = \lambda u$. Como M es una matriz no singular, tenemos que

$$\begin{aligned} (I - M^{-1}A)u &= \lambda u \\ M^{-1}Au &= u - \lambda u \\ MM^{-1}Au &= M(u - \lambda u) \\ Au &= (1-\lambda)Mu \end{aligned}$$

Note que $\lambda = 1$ si y sólo si $Au = 0$.
Como por hipótesis, A es definida positiva y u es cualquier vector propio de B, esto implica que $Au \neq 0$, por lo que $\lambda \neq 1$.
Luego

$$\begin{aligned} Au &= (1-\lambda)Mu \\ u^TAu &= (1-\lambda)u^TMu \\ \frac{1}{1-\lambda} &= \frac{u^TMu}{u^TAu}. \end{aligned}$$

Notemos que la transpuesta de la expresión anterior es

$$\frac{1}{1-\lambda} = \frac{u^TM^Tu}{u^TAu}.$$

De donde se sigue que

$$\begin{aligned} \frac{1}{1-\lambda} + \frac{1}{1-\lambda} &= \frac{u^TMu}{u^TAu} + \frac{u^TM^Tu}{u^TAu} \\ \frac{2(1-\lambda)}{(1-\lambda)^2} &= \frac{u^T(M+M^T)u}{u^TAu}. \end{aligned}$$

Por (2.9), tenemos que

$$\begin{aligned} \frac{2(1-\lambda)}{(1-\lambda)^2} &= \frac{u^T(Q+A)u}{u^TAu} \\ \frac{2(1-\lambda)}{(1-\lambda)^2} &= 1 + \frac{u^TQu}{u^TAu} \\ \frac{2(1-\lambda)}{(1-\lambda)^2} &> 1 \\ 2(1-\lambda) &> (1-\lambda)^2 \\ 2 - 2\lambda &> 1 - 2\lambda + \lambda^2 \\ 1 &> \lambda^2. \end{aligned}$$

De donde $0 \leqslant \lambda < 1$ y esto implica que $\rho(B) < 1$, por lo que se sigue del lema 1.1.3 que B es convergente.
Supongamos ahora que B es convergente y veamos que A es definida positiva. Sea $v_0 \in \mathbb{R}^n$ distinto de cero y consideramos la sucesión $\{v_p\}_{p\geq 0}$ definida por

$$v_{p+1} = Bv_p, \qquad p = 0, 1, 2, \ldots$$

Afirmamos que

$$v_p^T A v_p - v_{p+1}^T A v_{p+1} = (v_p - v_{p+1})^T Q (v_p - v_{p+1}), \quad p \geq 0. \tag{2.10}$$

En efecto, dado que $B = I - M^{-1}A$, entonces

$$\begin{aligned} v_p^T A v_p - v_{p+1}^T A v_{p+1} &= v_p^T A v_p - (Bv_p)^T A (Bv_p) \\ &= v_p^T A v_p - v_p^T B^T A B v_p \\ &= v_p^T (A - B^T A B) v_p \\ &= v_p^T (A - (I - M^{-1}A)^T A (I - M^{-1}A)) v_p. \end{aligned}$$

Pero

$$\begin{aligned} A - (I - M^{-1}A)^T A (I - M^{-1}A) &= A - \left[(I - A(M^{-1})^T) A (I - M^{-1}A)\right] \\ &= A - \left[(A - A(M^T)^{-1}A)(I - M^{-1}A)\right] \\ &= A - \left[(A - A(M^T)^{-1}A) - (A - A(M^T)^{-1}A)(M^{-1}A)\right] \\ &= A - \left[A - A(M^T)^{-1}A - (AM^{-1}A - A(M^T)^{-1}AM^{-1}A)\right] \\ &= A - \left[A - A(M^T)^{-1}A - AM^{-1}A + A(M^T)^{-1}AM^{-1}A\right] \\ &= A - A + A(M^T)^{-1}A + AM^{-1}A - A(M^T)^{-1}AM^{-1}A \\ &= A(M^T)^{-1}A + AM^{-1}A - A(M^T)^{-1}AM^{-1}A. \end{aligned}$$

Así que

$$v_p^T A v_p - v_{p+1}^T A v_{p+1} = v_p^T (A(M^T)^{-1}A + AM^{-1}A - A(M^T)^{-1}AM^{-1}A) v_p. \tag{2.11}$$

Por otro lado, notamos que

$$\begin{aligned} (v_p - v_{p+1})^T Q (v_p - v_{p+1}) &= (v_p^T - v_{p+1}^T)(Qv_p - Qv_{p+1}) \\ &= v_p^T (Qv_p - Qv_{p+1}) - v_{p+1}^T (Qv_p - Qv_{p+1}) \\ &= v_p^T Q v_p - v_p^T Q v_{p+1} - v_{p+1}^T Q v_p + v_{p+1}^T Q v_{p+1} \\ &= v_p^T Q v_p - v_p^T Q (Bv_p) - (Bv_p)^T Q v_p + (Bv_p)^T Q (Bv_p) \\ &= v_p^T Q v_p - v_p^T Q B v_p - v_p^T B^T Q v_p + v_p^T B^T Q B v_p \\ &= v_p^T (Q - QB - B^T Q + B^T Q B) v_p. \end{aligned}$$

Puesto que $Q = M + M^T - A$ y $B = I - M^{-1}A$, entonces

$$\begin{aligned} QB &= (M + M^T - A)(I - M^{-1}A) \\ &= M + M^T - A - (M + M^T - A)M^{-1}A \\ &= M + M^T - A - MM^{-1}A - M^TM^{-1}A + AM^{-1}A \\ &= M + M^T - A - A - M^TM^{-1}A + AM^{-1}A \\ &= M + M^T - M^TM^{-1}A + AM^{-1}A - 2A, \end{aligned} \tag{2.12}$$

$$\begin{aligned} B^TQ &= (I - M^{-1}A)^T(M + M^T - A) \\ &= (I - A(M^{-1})^T)(M + M^T - A) \\ &= M + M^T - A - A(M^T)^{-1}(M + M^T - A) \\ &= M + M^T - A - A(M^T)^{-1}M - A(M^T)^{-1}M^T + A(M^T)^{-1}A \\ &= M + M^T - A - A(M^T)^{-1}M - A + A(M^T)^{-1}A \\ &= M + M^T - A(M^T)^{-1}M + A(M^T)^{-1}A - 2A \end{aligned} \tag{2.13}$$

y

$$\begin{aligned} B^TQB &= (M + M^T - A(M^T)^{-1}M + A(M^T)^{-1}A - 2A)(I - M^{-1}A) \\ &= M + M^T - A(M^T)^{-1}M + A(M^T)^{-1}A - 2A - MM^{-1}A - M^TM^{-1}A \\ &\quad +A(M^T)^{-1}MM^{-1}A - A(M^T)^{-1}AM^{-1}A + 2AM^{-1}A \\ &= M + M^T - A(M^T)^{-1}M + A(M^T)^{-1}A - 2A - A - M^TM^{-1}A \\ &\quad +A(M^T)^{-1}A - A(M^T)^{-1}AM^{-1}A + 2AM^{-1}A \\ &= M + M^T - A(M^T)^{-1}M + A(M^T)^{-1}A - M^TM^{-1}A + A(M^T)^{-1}A \\ &\quad -A(M^T)^{-1}AM^{-1}A + 2AM^{-1}A - 3A. \end{aligned} \tag{2.14}$$

De (2.12), (2.13) y (2.14), obtenemos

$$\begin{aligned} Q - QB - B^TQ + B^TQB &= M + M^T - A - M - M^T + M^TM^{-1}A - AM^{-1}A + 2A \\ &\quad -M - M^T + A(M^T)^{-1}M - A(M^T)^{-1}A + 2A \\ &\quad +M + M^T - A(M^T)^{-1}M + A(M^T)^{-1}A - M^TM^{-1}A \\ &\quad +A(M^T)^{-1}A - A(M^T)^{-1}AM^{-1}A + 2AM^{-1}A - 3A \\ &= AM^{-1}A + A(M^T)^{-1}A - A(M^T)^{-1}AM^{-1}A. \end{aligned}$$

Así

$$(v_p - v_{p+1})^TQ(v_p - v_{p+1}) = v_p^T(AM^{-1}A + A(M^T)^{-1}A - A(M^T)^{-1}AM^{-1}A)v_p. \tag{2.15}$$

Finalmente, de (2.11) y (2.15) se concluye que

$$v_p^TAv_p - v_{p+1}^TAv_{p+1} = (v_p - v_{p+1})^TQ(v_p - v_{p+1}),$$

como queríamos.
Ahora probaremos que $\{v_p^T A v_p\}_{p\geq 0}$ es no creciente. Como Q es definida positiva, entonces $v^T Q v > 0$ para todo $v \neq 0$, en particular, para $v_p - v_{p+1}$, $p \geq 0$, se sigue de (2.10) que

$$v_p^T A v_p - v_{p+1}^T A v_{p+1} = (v_p - v_{p+1})^T Q (v_p - v_{p+1}) \geq 0.$$

Por lo que $v_{p+1}^T A v_{p+1} \leq v_p^T A v_p$ para todo $p \geq 0$, es decir, $\{v_p^T A v_p\}_{p\geq 0}$ es no creciente.
Finalmente, probaremos que A es definida positiva. Supongamos que $v_0^T A v_0 \leq 0$ para algún $v_0 \neq 0$. Como B es convergente, entonces $v_1 \neq v_0$, por lo que

$$v_p^T A v_p \leq v_1^T A v_1 < v_0^T A v_0 \leq 0, \quad p \geq 2$$

lo que implica que

$$\lim_{p\to\infty} v_p^T A v_p \leq v_0^T A v_0 < 0,$$

el cual es una contradicción ya que la convergencia de B implica

$$\lim_{p\to\infty} v_p = 0.$$

Por lo tanto, $v_0^T A v_0 > 0$, es decir, A es definida positiva. □

Corolario 2.5.1 (Isaacson and Keller (1966)[4]). *Si $A \in \mathbb{R}^{n\times n}$ es simétrica con elementos positivos en la diagonal, entonces el método de Gauss-Seidel converge si y sólo si A es definida positiva.*

Demostración. Supongamos que A es definida positiva y consideramos la descomposición (2.1). Entonces

$$A = D - L - L^T,$$

donde D es la matriz diagonal con elementos de la diagonal de A que son positivos y L es la parte negativa triangular inferior estricta de A. Del método de Gauss-Seidel y de acuerdo con las descomposiciones (1.6) y (2.1), se sigue que

$$M = D - L \quad \text{y} \quad N = L^T.$$

Luego

$$Q \equiv M + M^T - A = D.$$

Es claro que M es no singular y como los valores propios de D son positivos, entonces Q es definida positiva. Así que se sigue del teorema 2.5.3 que $B = I - M^{-1}A$ es convergente si y sólo si es A definida positiva, es decir, el método de Gauss-Seidel es convergente si y sólo si A es definida positiva. □

Similarmente, tenemos un resultado sobre la convergencia del método de Jacobi como un caso especial del siguiente corolario.

Corolario 2.5.2 (Isaacson and Keller (1996)[4]). *Si D es simétrica y no singular y $D+L+L^T$ es definida positiva, entonces $D^{-1}(L+L^T)$ es convergente si y sólo si $A = D - L - L^T$ es definida positiva.*

Demostración. Aplicamos el teorema 2.5.3 para $M = D$. □

En el caso especial en que D es una matriz diagonal con elementos de la diagonal de A, el corolario 2.5.2 sostiene la convergencia del método de Jacobi para la matriz A. En este caso, el corolario 2.5.2 es equivalente a:

Corolario 2.5.3. *Si $A \in \mathbb{R}^{n\times n}$ es simétrica y si la matriz que resulta de sustituir los elementos a_{ij}, $i \neq j$, por sus opuestos, es definida positiva, entonces el método de Jacobi converge si y sólo si A es definida positiva.*

2.6. Métodos de relajación y su convergencia

Sea ω un número real no nulo y consideramos las descomposiciones (1.6) y (2.1) de la matriz A, dadas por $A = M - N = D - L - U$. Entonces la matriz A también se puede escribir como

$$A = \frac{1}{\omega}D - L - \frac{1-\omega}{\omega}D - U\,. \tag{2.16}$$

Si se eligen

$$M = \frac{1}{\omega}D - L \qquad \text{y} \qquad N = \frac{1-\omega}{\omega}D + U\,,$$

el método iterativo correspondiente será

$$x^{(k+1)} = \left(\frac{1}{\omega}D - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right)x^{(k)} + \left(\frac{1}{\omega}D - L\right)^{-1} b\,, \quad k \geq 0 \tag{2.17}$$

o bien

$$\left(\frac{1}{\omega}D - L\right)x^{(k+1)} = \left(\frac{1-\omega}{\omega}D + U\right)x^{(k)} + b\,, \quad k \geq 0\,, \tag{2.18}$$

o en términos de componentes

$$\left.\begin{aligned}
x_1^{(k+1)} &= \frac{\omega}{a_{11}}\left(b_1 + \frac{1-\omega}{\omega}a_{11}x_1^{(k)} - \sum_{j=2}^{n} a_{1j}x_j^{(k)}\right), \quad k \geq 0\,, \\
x_i^{(k+1)} &= \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \frac{1-\omega}{\omega}a_{ii}x_i^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \\
&\quad i = 2 : n-1, \;\; k \geq 0\,, \\
x_n^{(k+1)} &= \frac{\omega}{a_{nn}}\left(b_n + \frac{1-\omega}{\omega}a_{nn}x_n^{(k)} - \sum_{j=1}^{n-1} a_{nj}x_j^{(k)}\right), \quad k \geq 0\,,
\end{aligned}\right\} \tag{2.19}$$

bajo el supuesto que $a_{ii} \neq 0$, $i = 1 : n$.
La matriz de iteración del método de relajación, lo denotaremos por E_ω, y está dada por

$$E_\omega = \left(\frac{1}{\omega}D - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right), \quad \omega \neq 0. \tag{2.20}$$

El parámetro ω se llama de **relajación** y es claro que cuando vale 1, se tiene el método de Gauss-Seidel. Cuando $\omega > 1$, el método correspondiente se le llama de **sobrerelajación sucesiva**. Estos métodos de sobrerelajación sucesiva se abrevian como **SOR** (*Successive Over Relaxation*) y son particularmente útiles para resolver sistemas de ecuaciones lineales que se presentan en la solución numérica de ecuaciones diferenciales parciales [1], como lo veremos en el Capítulo 4.
Iniciamos la revisión de algunos resultados de convergencia sobre el método de relajación. El siguiente teorema muestra que el método de relajación converge solo con parámetros ω tales que $0 < \omega < 2$.

Teorema 2.6.1 (de Kahan, Stoer and Bulirsch (1993)[9]). *$\rho(E_\omega) \geq |\omega - 1|$ para todo $\omega \neq 0$.*

Demostración. De (2.20) tenemos que

$$\begin{aligned} E_\omega &= \left(\frac{1}{\omega}D - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right) \\ &= (D - \omega L)^{-1}((1-w)D + wU) \\ &= [D(I - \omega D^{-1}L)]^{-1}((1-w)D + wU) \\ &= (I - \omega D^{-1}L)^{-1}D^{-1}((1-w)D + wU) \\ &= (I - \omega D^{-1}L)^{-1}((1-w)I + wD^{-1}U). \end{aligned}$$

Luego

$$\det(E_\omega) = \det((I - \omega D^{-1}L)^{-1})\det((1-w)I + wD^{-1}U).$$

Como $D^{-1}L$ es una matriz triangular inferior con ceros en la diagonal principal, entonces $I - \omega D^{-1}L$ sigue siendo una matriz triangular inferior con unos en la diagonal principal, por consiguiente su inversa es también triangular inferior con unos en la diagonal principal, así

$$\det((I - \omega D^{-1}L)^{-1}) = 1.$$

Por otra parte, $(1-w)I + wD^{-1}U$ es a su vez una matriz triangular superior con $1 - \omega$ en su diagonal principal. Así

$$\det((1-w)I + wD^{-1}U) = (1-\omega)^n.$$

Ahora considerando que el determinante de una matriz es el producto de sus valores propios y denotando por λ_i a los valores propios de E_ω, tenemos que

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

De la definición 1.1.3 tenemos

$$\rho(E_\omega) \geq |\lambda_i|, \quad i = 1:n.$$

De esta manera

$$[\rho(E_\omega)]^n \geq \prod_{i=1}^{n} |\lambda_i| = |1-\omega|^n.$$

Así tomando raíz $n-$ésima, resulta la desigualdad $\rho(E_\omega) \geq |1-\omega|$. □

El Teorema 2.6.1 dice que una condición necesaria para la convergencia del método de relajación es que $|\omega - 1| < 1$, y esto significa que $0 < \omega < 2$.

Teorema 2.6.2 (Ostrowski and Reich, Vargas (1962)[11]). *Sea $A = D - L - L^T \in \mathbb{R}^{n\times n}$ simétrica, donde D es definida positiva y $D - \omega L$ es no singular para $0 < \omega < 2$. Entonces, $\rho(E_\omega) < 1$ si y sólo si A es definida positiva y $0 < \omega < 2$.*

Demostración. Consideramos por hipótesis que $A = D - L - L^T$. Si ε_0 es un vector de error inicial arbitrario distinto de cero, entonces los vectores de errores sucesivos para el método iterativo de relajación están definidos por

$$\varepsilon_{m+1} = E_\omega \varepsilon_m, \quad m \geq 0, \tag{2.21}$$

o equivalentemente

$$(D - \omega L)\varepsilon_{m+1} = ((1-\omega)D + \omega L^T)\varepsilon_m, \quad m \geq 0. \tag{2.22}$$

Desarrollamos ambos miembros de (2.22):

$$\begin{aligned}
D\varepsilon_{m+1} - \omega L\varepsilon_{m+1} &= \omega L^T\varepsilon_m + D\varepsilon_m - \omega D\varepsilon_m \\
D\varepsilon_{m+1} - D\varepsilon_m - \omega L\varepsilon_{m+1} &= \omega L^T\varepsilon_m - \omega D\varepsilon_m \\
D\varepsilon_{m+1} - D\varepsilon_m - \omega L\varepsilon_{m+1} + \omega L\varepsilon_m &= \omega L\varepsilon_m + \omega L^T\varepsilon_m - \omega D\varepsilon_m \\
D\varepsilon_m - D\varepsilon_{m+1} - \omega L\varepsilon_m + \omega L\varepsilon_{m+1} &= \omega D\varepsilon_m - \omega L\varepsilon_m - \omega L^T\varepsilon_m
\end{aligned}$$

$$\begin{aligned}
D(\varepsilon_m - \varepsilon_{m+1}) - \omega L(\varepsilon_m - \varepsilon_{m+1}) &= \omega(D - L - L^T)\varepsilon_m \\
(D - \omega L)(\varepsilon_m - \varepsilon_{m+1}) &= \omega A\varepsilon_m, \quad m \geq 0.
\end{aligned}$$

Sea $\delta_m := \varepsilon_m - \varepsilon_{m+1}$, $m \geq 0$. Se sigue de la igualdad anterior que

$$(D - \omega L)\delta_m = \omega A\varepsilon_m, \quad m \geq 0. \tag{2.23}$$

De (2.22) también tenemos

$$\begin{aligned}
D\varepsilon_{m+1} - \omega L\varepsilon_{m+1} &= (1-\omega)D\varepsilon_m + \omega L^T\varepsilon_m \\
\omega D\varepsilon_{m+1} - \omega L\varepsilon_{m+1} &= (1-\omega)D\varepsilon_m + \omega L^T\varepsilon_m - D\varepsilon_{m+1} + \omega D\varepsilon_{m+1} \\
\omega D\varepsilon_{m+1} - \omega L\varepsilon_{m+1} - \omega L^T\varepsilon_{m+1} &= (1-\omega)D\varepsilon_m + \omega L^T\varepsilon_m - (1-\omega)D\varepsilon_{m+1} - \omega L^T\varepsilon_{m+1} \\
\omega(D - L - L^T)\varepsilon_{m+1} &= (1-\omega)D\varepsilon_m - (1-\omega)D\varepsilon_{m+1} + \omega L^T\varepsilon_m - \omega L^T\varepsilon_{m+1} \\
\omega A\varepsilon_{m+1} &= (1-\omega)D(\varepsilon_m - \varepsilon_{m+1}) + \omega L^T(\varepsilon_m - \varepsilon_{m+1}) \\
\omega A\varepsilon_{m+1} &= (1-\omega)D\delta_m + \omega L^T\delta_m, \quad m \geq 0. \qquad (2.24)
\end{aligned}$$

Si multiplicamos a la izquierda de (2.23) y (2.24), respectivamente, por ε_m^T y ε_{m+1}^T, obtenemos

$$\omega\varepsilon_m^T A\varepsilon_m = \varepsilon_m^T D\delta_m - \omega\varepsilon_m^T L\delta_m, \quad m \geq 0,$$

y

$$\omega\varepsilon_{m+1}^T A\varepsilon_{m+1} = (1-\omega)\varepsilon_{m+1}^T D\delta_m + \omega\varepsilon_{m+1}^T L^T\delta_m, \quad m \geq 0.$$

Restando la última igualdad de la igualdad anterior, y haciendo uso de que A es simétrica y aplicando (2.23), tenemos:

$$\begin{aligned}
\omega(\varepsilon_m^T A\varepsilon_m - \varepsilon_{m+1}^T A\varepsilon_{m+1}) &= \varepsilon_m^T D\delta_m - \omega\varepsilon_m^T L\delta_m \\
&\quad -\varepsilon_{m+1}^T D\delta_m + \omega\varepsilon_{m+1}^T D\delta_m - \omega\varepsilon_{m+1}^T L^T\delta_m \\
&= \delta_m^T D\delta_m - \omega\varepsilon_m^T L\delta_m + \omega\varepsilon_{m+1}^T D\delta_m - \omega\varepsilon_{m+1}^T L^T\delta_m \\
&= \delta_m^T D\delta_m - \omega\varepsilon_m^T L\delta_m + \omega(\varepsilon_m^T - \delta_m^T)D\delta_m - \omega(\varepsilon_m^T - \delta_m^T)L^T\delta_m \\
&= \delta_m^T D\delta_m - \omega\varepsilon_m^T L\delta_m + \omega\varepsilon_m^T D\delta_m - \omega\delta_m^T D\delta_m \\
&\quad -\omega\varepsilon_m^T L^T\delta_m + \omega\delta_m^T L^T\delta_m \\
&= \delta_m^T D\delta_m + \omega\varepsilon_m^T(D - L - L^T)\delta_m - \omega\delta_m^T D\delta_m + \omega\delta_m^T L^T\delta_m \\
&= \delta_m^T D\delta_m + \omega\varepsilon_m^T A\delta_m - \omega\delta_m^T D\delta_m + \omega\delta_m^T L^T\delta_m \\
&= \delta_m^T D\delta_m + (\omega A\varepsilon_m)^T\delta_m - \omega\delta_m^T D\delta_m + \omega\delta_m^T L^T\delta_m \\
&= \delta_m^T D\delta_m + [(D - \omega L)\delta_m]^T\delta_m - \omega\delta_m^T D\delta_m + \omega\delta_m^T L^T\delta_m \\
&= \delta_m^T D\delta_m + \delta_m^T(D - \omega L^T)\delta_m - \omega\delta_m^T D\delta_m + \omega\delta_m^T L^T\delta_m \\
&= 2\delta_m^T D\delta_m - \omega\delta_m^T L^T\delta_m - \omega\delta_m^T D\delta_m + \omega\delta_m^T L^T\delta_m \\
&= (2-\omega)\delta_m^T D\delta_m.
\end{aligned}$$

Por lo tanto,

$$(2-\omega)\delta_m^T D\delta_m = \omega(\varepsilon_m^T A\varepsilon_m - \varepsilon_{m+1}^T A\varepsilon_{m+1}), \quad m \geq 0. \qquad (2.25)$$

Supongamos ahora que A es definida positiva y $0 < \omega < 2$. Sea λ un valor propio arbitrario de E_ω y escogemos ε_0 como un vector propio de E_ω, con valor propio λ. Entonces $\varepsilon_1 := E_\omega \varepsilon_0 = \lambda \varepsilon_0$, de donde $\delta_0 = (1-\lambda)\varepsilon_0$. Con esta particular elección de ε_0, (2.25) se reduce a

$$\left(\frac{2-\omega}{\omega}\right)(1-\lambda)^2 \varepsilon_0^T D \varepsilon_0 = (1-\lambda^2)\varepsilon_0^T A \varepsilon_0. \tag{2.26}$$

Ahora, λ no puede ser uno, porque entonces δ_0 debería ser el vector nulo y por (2.23), $A\varepsilon_0 = 0$. Como ε_0 es por definición distinto de cero, esto contradice el supuesto que A es definida positiva. Como $0 < \omega < 2$, el lado izquierdo de (2.26) es positivo y como $\varepsilon_0^T A \varepsilon_0 > 0$, entonces $\lambda^2 < 1$ lo que implica $|\lambda| < 1$. Así que $\rho(E_\omega) < 1$.
Ahora, supongamos que $\rho(E_\omega) < 1$. Se sigue del teorema 2.6.1 que $0 < \omega < 2$. Para cualquier vector ε_0, $\lim_{m\to\infty} \varepsilon_m = 0$ y esto implica $\lim_{m\to\infty} \varepsilon_m^T A \varepsilon_m = 0$. Como D es definida positiva, entonces $\delta_m^T D \delta_m > 0$ y de (2.25) se sigue que

$$\varepsilon_m^T A \varepsilon_m = \varepsilon_{m+1}^T A \varepsilon_{m+1} + \left(\frac{2-\omega}{\omega}\right)\delta_m^T D \delta_m \geq \varepsilon_{m+1}^T A \varepsilon_{m+1}, \quad m \geq 0. \tag{2.27}$$

Si A no fuera definida positiva, podemos encontrar un vector ε_0 tal que $\varepsilon_0^T A \varepsilon_0 \leq 0$. Por otro lado, los valores propios de E_ω no pueden ser uno, así $\delta_0 = \varepsilon_0 - \varepsilon_1 = \varepsilon_0 - E_\omega \varepsilon_0 = (I - E_\omega)\varepsilon_0$ es un vector distinto de cero, por lo que $\delta_0^T D \delta_0 > 0$, y (2.27) nos da la desigualdad estricta $\varepsilon_1^T A \varepsilon_1 < \varepsilon_0^T A \varepsilon_0 \leq 0$. La propiedad no creciente de (2.27) ahora contradice que $\lim_{m\to\infty} \varepsilon_m^T A \varepsilon_m = 0$. Esto prueba que A debe ser definida positiva. □

Existe una clase importante de matrices donde las afirmaciones cualitativas de los teoremas 2.6.1 y 2.6.2, se pueden agudizar considerablemente. Esta clase de matrices son las que tienen la *propiedad A* que la introdujo Young (1971) [12] y es generalizada por Varga (1962) [11] a la clase de matrices *coherentemente ordenadas*, [8, 9].

Definición 2.6.1. *Una matriz $A \in \mathbb{R}^{n\times n}$ tiene la propiedad A si existe una matriz de permutación $P \in \mathbb{R}^{n\times n}$ tal que PAP^T tiene la forma*

$$PAP^T = \begin{bmatrix} D_1 & M_1 \\ M_2 & D_2 \end{bmatrix}, \tag{2.28}$$

donde D_1 y D_2 son matrices diagonales.

La razón de ser de las matrices con propiedad A la enunciamos en el siguiente teorema.

Teorema 2.6.3 (Stoer and Bulirsch (1993) [9]). *Para matrices $A \in \mathbb{R}^{n\times n}$ con propiedad A y $a_{ii} \neq 0$, $i = 1 : n$, existe una matriz de permutación $P \in \mathbb{R}^{n\times n}$ tal que la descomposición (2.1) de $\overline{A} = D - L - U$ de la matriz permutada $\overline{A} = PAP^T$ tiene la siguiente propiedad: los valores propios de la matriz*

$$J(\alpha) = \alpha D^{-1} L + \alpha^{-1} D^{-1} U, \quad \alpha \in \mathbb{C}, \quad \alpha \neq 0,$$

son independientes de α.

Demostración. Por la definición 2.6.1, existe una matriz de permutación $P \in \mathbb{R}^{n\times n}$ tal que

$$PAP^T = \begin{bmatrix} D_1 & M_1 \\ M_2 & D_2 \end{bmatrix} = D - L - U = D(I - D^{-1}L - D^{-1}U).$$

Así que

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \quad D^{-1}L = -\begin{bmatrix} 0 & 0 \\ D_2^{-1}M_2 & 0 \end{bmatrix} \quad \text{y} \quad D^{-1}U = -\begin{bmatrix} 0 & D_1^{-1}M_1 \\ 0 & 0 \end{bmatrix},$$

donde D_1 y D_2 son matrices diagonales no singulares. Para $\alpha \neq 0$, tenemos ahora que

$$\begin{aligned} J(\alpha) &= -\begin{bmatrix} 0 & \alpha^{-1}D_1^{-1}M_1 \\ \alpha D_2^{-1}M_2 & 0 \end{bmatrix} \\ &= -S_\alpha \begin{bmatrix} 0 & D_1^{-1}M_1 \\ D_2^{-1}M_2 & 0 \end{bmatrix} S_\alpha^{-1} \\ &= S_\alpha J(1) S_\alpha^{-1}, \end{aligned}$$

donde S_α es la matriz diagonal no singular

$$S_\alpha = \begin{bmatrix} I_1 & 0 \\ 0 & \alpha I_2 \end{bmatrix},$$

donde a su vez I_1 e I_2 son matrices identidad (no necesariamente del mismo tamaño).
Así que las matrices $J(\alpha)$ y $J(1)$ son similares o semejantes, por lo que tienen los mismos valores propios. Esto prueba que los valores propios de $J(\alpha)$ son independientes de α. □

Siguiendo a Varga (1962) [11], tenemos la siguiente definición.

Definición 2.6.2. *Una matriz $A \in \mathbb{R}^{n\times n}$, con la descomposición* (2.1) *de $A = D - L - U$, tiene la propiedad que los valores propios de matrices*

$$J(\alpha) = \alpha D^{-1}L + \alpha^{-1}D^{-1}U, \quad \alpha \neq 0, \tag{2.29}$$

son independientes de α, se llama ***coherentemente ordenada*** *(*consistently ordered *en inglés).*

El teorema 2.6.3 afirma que las matrices con propiedad A son coherentemente ordenadas, es decir, las filas y columnas de una matriz A con propiedad A se pueden reagrupar mediante una permutación P tal que PAP^T sea coherentemente ordenada.
Sin embargo, existen matrices coherentemente ordenadas que no tienen la propiedad A, como se muestra en el ejemplo 2.6.4.

Ejemplo 2.6.1 (Skiva (2005) [8])**.** *Las matrices tridiagonales $n \times n$ con elementos diagonales no nulos son coherentemente ordenadas.*

Demostración. Sea la matriz tridiagonal

$$A = \begin{bmatrix} a_1 & c_1 & & \\ b_2 & a_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & b_n & a_n \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad a_i \neq 0,\ i = 1:n.$$

Sea $A = D - L - U$ la descomposición dada en (2.1), donde

$$D = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_n \end{bmatrix},\ L = -\begin{bmatrix} 0 & 0 & & \\ b_2 & 0 & \ddots & \\ & \ddots & \ddots & 0 \\ & & b_n & 0 \end{bmatrix},\ U = -\begin{bmatrix} 0 & c_1 & & \\ 0 & 0 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & 0 & 0 \end{bmatrix}.$$

Notemos que

$$J(\alpha) = \alpha D^{-1}L + \alpha^{-1}D^{-1}U = -\begin{bmatrix} 0 & \alpha^{-1}\dfrac{c_1}{a_1} & & \\ \alpha\dfrac{b_2}{a_2} & 0 & \ddots & \\ & \ddots & \ddots & \alpha^{-1}\dfrac{c_{n-1}}{a_{n-1}} \\ & & \alpha\dfrac{b_n}{a_n} & 0 \end{bmatrix}, \quad \alpha \neq 0.$$

Es claro que $S_\alpha = \operatorname{diag}(1, \alpha, \ldots, \alpha^{n-1})$ es no singular para todo $\alpha \neq 0$, y satisface la relación

$$J(\alpha) = S_\alpha J(1) S_\alpha^{-1}.$$

Así que $J(\alpha)$ y $J(1)$ son semejantes, por lo que tienen los mismos valores propios. Pero esto prueba que justamente, los valores propios de $J(\alpha)$ son independientes de α. Así que la matriz tridiagonal A con elementos no nulos en la diagonal es coherentemente ordenada. □

Ejemplo 2.6.2 (Stoer and Bulirsch (1993) [9]). *Las matrices tridiagonales por bloques de la forma*

$$A = \begin{bmatrix} D_1 & A_{12} & & & \\ A_{21} & D_2 & A_{23} & & \\ & \ddots & \ddots & \ddots & \\ & & A_{N-1,N-2} & D_{N-1} & A_{N-1,N} \\ & & & A_{N,N-1} & D_N \end{bmatrix}$$

con matrices diagonales no singulares D_i, $i = 1:N$, son coherentemente ordenadas.

Demostración. Si todas las matrices diagonales D_i son no singulares, entonces para cualquier $\alpha \neq 0$, la matriz

$$J(\alpha) = -\begin{bmatrix} 0 & \alpha^{-1}D_1^{-1}A_{12} & & & \\ \alpha D_2^{-1}A_{21} & 0 & \alpha^{-1}D_2^{-1}A_{23} & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha D_{N-1}^{-1}A_{N-1,N-21} & 0 & \alpha^{-1}D_{N-1}^{-1}A_{N-1,N} \\ & & & \alpha D_N^{-1}A_{N,N-1} & 0 \end{bmatrix}$$

satisface la relación

$$J(\alpha) = S_\alpha J(1) S_\alpha^{-1}, \quad S_\alpha := \begin{bmatrix} I_1 & & & \\ & \alpha I_2 & & \\ & & \ddots & \\ & & & \alpha^{N-1} I_N \end{bmatrix} \quad \forall \alpha \neq 0.$$

Por lo que $J(\alpha)$ y $J(1)$ son semejantes, así que tienen los mismos valores propios. Pero esto prueba que justamente, los valores propios de $J(\alpha)$ son independientes de α. Así que la matriz A tridiagonal por bloques con matrices diagonales no singulares es coherentemente ordenada. □

De hecho, cualquier matriz tridiagonal por bloques también es coherentemente ordenada [8].

Ejemplo 2.6.3 (Stoer and Bulirsch (1993) [9])**.** *Las matrices tridiagonales por bloques tienen la propiedad A.*

Demostración. Mostraremos solo para el caso especial de la siguiente matriz 3×3:

$$A = \begin{bmatrix} 1 & b & 0 \\ a & 1 & d \\ 0 & c & 1 \end{bmatrix}.$$

Notemos que

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

es una matriz de permutación, y puesto que

$$PAP^T = \left[\begin{array}{c|cc} 1 & a & d \\ \hline b & 1 & 0 \\ c & 0 & 1 \end{array}\right]$$

tiene la forma (2.28), entonces la matriz tridiagonal A tiene la propiedad A. El caso general, se procede análogamente. □

Ejemplo 2.6.4 (Stoer and Bulirsch (1993) [9]). *Existen matrices coherentemente ordenadas que no tienen la propiedad A.*

Demostración. Basta exhibir un ejemplo. La matriz triangular inferior

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

es coherentemente ordenada pero no tiene la propiedad A. Primero mostraremos que A es coherentemente ordenada. Sea $A = D - L - U$ la descomposición dada en (2.1), donde

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; L = -\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \; U = -\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Notemos que

$$\begin{aligned} J(\alpha) &= \alpha D^{-1}L + \alpha^{-1}D^{-1}U, \quad \alpha \neq 0 \\ &= -\alpha \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} + \alpha^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= -\alpha \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \\ &= \alpha L \end{aligned}$$

Luego, observe que el polinomio característico de L es

$$\begin{aligned} p(\lambda) &= \left| \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \right| \\ &= \begin{vmatrix} -\lambda & 0 & 0 \\ -1 & -\lambda & 0 \\ -1 & -1 & -\lambda \end{vmatrix} \\ &= -\lambda^3. \end{aligned}$$

Entonces los valores propios de L son $\lambda_1 = \lambda_2 = \lambda_3 = 0$. De igual manera, el polinomio

característico de αL es

$$\begin{aligned} p(\lambda) &= \left| \begin{bmatrix} 0 & 0 & 0 \\ -\alpha & 0 & 0 \\ -\alpha & -\alpha & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \right| \\ &= \begin{vmatrix} -\lambda & 0 & 0 \\ -\alpha & -\lambda & 0 \\ -\alpha & -\alpha & -\lambda \end{vmatrix} \\ &= -\lambda^3. \end{aligned}$$

Por lo que, los valores propios de αL también son $\lambda_1 = \lambda_2 = \lambda_3 = 0$. Esto quiere decir que los valores propios de $J(\alpha)$ son independientes de α. Así que la matriz A es coherentemente ordenada.

Ahora probaremos que A no tiene la propiedad A. Puesto que todas las posibles matrices de permutación en $\mathbb{R}^{3\times 3}$ son:

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$P_4 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad P_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ y } \quad P_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Y ninguna de las matrices

$$P_1AP_1^T = A, \qquad P_2AP_2^T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad P_3AP_3^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P_4AP_4^T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_5AP_5^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad P_6AP_6^T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix},$$

tienen la forma (2.28), por lo que la matriz A no tiene la propiedad A. □

La importancia de las matrices coherentemente ordenadas (y por ende del teorema 2.6.3, e indirectamente de las matrices con propiedad A) radica en el hecho de que se puede mostrar explícitamente que los valores propios μ de $E_J = D^{-1}L + D^{-1}U$ están relacionados con los valores propios $\lambda = \lambda(\omega)$ de $E_\omega = (D-\omega L)^{-1}((1-\omega)D+\omega U)$, para una matriz $A = D-L-U$ como en (2.1). Esto se muestra en el siguiente teorema.

Teorema 2.6.4 (Young, Varga, [9]). *Sea $A \in \mathbb{R}^{n\times n}$ coherentemente ordenada y $\omega \neq 0$. Entonces*

a) *Si μ es un valor propio de $E_J = D^{-1}L + D^{-1}U$, entonces $-\mu$ también es un valor propio de E_J.*

b) *Si μ es un valor propio de E_J y*

$$(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2, \tag{2.30}$$

entonces λ es un valor propio de E_ω.

c) *Si $\lambda \neq 0$ es un valor propio de E_ω y (2.30) se satisface, entonces μ es un valor propio de E_J.*

Demostración. a) Consideramos $J(\alpha)$ definida en (2.29). Puesto que A es coherentemente ordenada, las matrices $J(-1) = -D^{-1}L - D^{-1}U = -E_J$ y $J(1) = D^{-1}L + D^{-1}U = E_J$ tienen los mismos valores propios. Así, si μ es un valor propio de E_J, entonces μ es un valor propio de $-E_J$, por lo que

$$\begin{aligned} \det(\mu I + E_J) = 0 &\implies \det(-(-\mu I - E_J)) = 0 \\ &\implies (-)^n \det(-\mu I - E_J) = 0 \\ &\implies \det(-\mu I - E_J) = 0, \end{aligned}$$

lo que implica que $-\mu$ es valor propio de E_J.

b) Notemos que

$$E_\omega = (D - \omega L)^{-1}((1-\omega)D + \omega U) = (I - \omega D^{-1}L)^{-1}((1-\omega)I + \omega D^{-1}U),$$

y que $\det(I - \omega D^{-1}L) = 1$ para todo ω.
Así que

$$\begin{aligned} \det(\lambda I - E_\omega) &= \det((I - \omega D^{-1}L)(\lambda I - E_\omega)) \\ &= \det((I - \omega D^{-1}L)(\lambda I - (I - \omega D^{-1}L)^{-1}((1-\omega)I + \omega D^{-1}U))) \\ &= \det(\lambda I - \lambda\omega D^{-1}L - (1-\omega)I - \omega D^{-1}U)) \\ &= \det((\lambda + \omega - 1)I - \lambda\omega D^{-1}L - \omega D^{-1}U)). \end{aligned} \tag{2.31}$$

Sea ahora μ un valor propio de $E_J = D^{-1}L + D^{-1}U$ y λ una solución de (2.30). Entonces $\lambda + \omega - 1 = \pm\sqrt{\lambda}\omega\mu$. Por (a) podemos suponer sin pérdida de generalidad que

$$\lambda + \omega - 1 = \sqrt{\lambda}\omega\mu.$$

Si $\lambda = 0$, entonces $\omega = 1$, y por (2.31),

$$\det(0 \cdot I - E_1) = \det(-D^{-1}U) = 0,$$

lo que significa que $\lambda = 0$ es un valor propio de E_1. Si $\lambda \neq 0$, se sigue de (2.31) que

$$\begin{aligned}
\det(\lambda I - E_\omega) &= \det\left((\lambda+\omega-1)I - \sqrt{\lambda}\omega\left(\sqrt{\lambda}D^{-1}L + \frac{1}{\sqrt{\lambda}}D^{-1}U\right)\right) \\
&= \det\left(\sqrt{\lambda}\omega\mu I - \sqrt{\lambda}\omega\left(\sqrt{\lambda}D^{-1}L + \frac{1}{\sqrt{\lambda}}D^{-1}U\right)\right) \\
&= \left(\sqrt{\lambda}\omega\right)^n \det\left(\mu I - \left(\sqrt{\lambda}D^{-1}L + \frac{1}{\sqrt{\lambda}}D^{-1}U\right)\right) \\
&= \left(\sqrt{\lambda}\omega\right)^n \det\left(\mu I - \left(D^{-1}L + D^{-1}U\right)\right) \\
&= \left(\sqrt{\lambda}\omega\right)^n \det\left(\mu I - E_J\right) \qquad (2.32) \\
&= 0.
\end{aligned}$$

Puesto que los valores propios de las matrices $J(\sqrt{\lambda}) = \sqrt{\lambda}D^{-1}L + \frac{1}{\sqrt{\lambda}}D^{-1}U$ son los mismos valores propios de $J(1) = D^{-1}L + D^{-1}U = E_J$ y μ es un valor propio de E_J, por lo tanto, $\det(\lambda I - E_\omega) = 0$, de donde λ es un valor propio de E_ω.

c) Recíprocamente, sea $\lambda \neq 0$ un valor propio de E_ω, y μ un número que satisface (2.30), es decir, $\lambda + \omega - 1 = \pm\omega\sqrt{\lambda}\mu$. En vista de (a) es suficiente mostrar que el número μ con $\lambda + \omega - 1 = \omega\sqrt{\lambda}\mu$ es un valor propio de E_J. Pero esto se sigue inmediatamente de (2.32). □

El siguiente resultado se sigue del teorema 2.6.4 con $\omega = 1$:

Corolario 2.6.1. *Si $A \in \mathbb{R}^{n\times n}$ es coherentemente ordenada, entonces la matriz de iteración $E_{GS} = (D-L)^{-1}U$ del método de Gauss-Seidel satisface*

$$\rho(E_{GS}) = [\rho(E_J)]^2,$$

donde E_J es la matriz de iteración del método de Jacobi.

El corolario afirma que para matrices coherentemente ordenadas, el método de Gauss-Seidel converge dos veces más rápido que el método de Jacobi.

Exhibiremos ahora un caso especial importante del parámetro de relajación óptimo ω_b, caracterizado por el teorema de Kahan 2.6.1:

$$\rho(E_{\omega_b}) = \min_{\omega\in\mathbb{R}} \rho(E_\omega) = \min_{0<\omega<2} \rho(E_\omega).$$

Teorema 2.6.5 (Young, Varga, [9])**.** *Sea $A \in \mathbb{R}^{n\times n}$ coherentemente ordenada tal que $A = D - L - U$ como en (2.1), y que los valores propios de $E_J = D^{-1}(L+U)$ son reales y $\rho(E_J) < 1$. Entonces*

$$\omega_b = \frac{2}{1+\sqrt{1-[\rho(E_J)]^2}} \quad \text{y} \quad \rho(E_{\omega_b}) = \omega_b - 1 = \left(\frac{\rho(E_J)}{1+\sqrt{1-[\rho(E_J)]^2}}\right)^2. \qquad (2.33)$$

En general,

$$\rho(E_\omega) = \begin{cases} \omega - 1 & \text{si } \omega_b < \omega < 2 \\ 1 - \omega + \frac{1}{2}\mu^2\omega^2 + \mu\omega\sqrt{1 - \omega + \frac{1}{4}\mu^2\omega^2} & \text{si } 0 < \omega \leq \omega_b, \end{cases} \tag{2.34}$$

donde $\mu = \rho(E_J)$.

Demostración. Consultarlo en el libro de Stoer and Bulirsch (1993) [9]. □

Ejemplo 2.6.5. *En la figura 2.1 se muestra la gráfica del radio espectral* $\rho(E_\omega)$ *dado en* (2.34), *de la matriz tridiagonal*

$$A = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{100\times 100}.$$

Por (2.33) *notemos que* $\omega_b = 1.9397$.

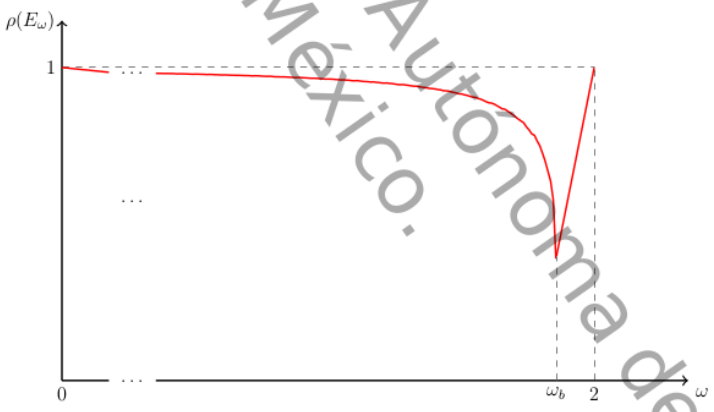


Figura 2.1: Gráfica del radio espectral $\rho(E_\omega)$ del método de relajación, de la matriz tridiagonal A del ejemplo 2.6.5. Recuerde que $\omega_b = 1.9397$.

Ejemplo 2.6.6. *Calcular las primeras tres aproximaciones de la solución de* $Ax = b$ *usando el método de relajación, donde* A *y* b *están dadas en el ejemplo 2.1.1, con el punto de inicio*

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad y \quad \omega = \frac{1}{2}.$$

Solución. Primero factorizamos la matriz A como $A = D - L - U$, donde D, L y U son como en el ejemplo 2.1.1. Para aplicar (2.17), calculamos $\left(\frac{1}{\omega}D - L\right)^{-1}$, $\left(\frac{1}{\omega}D - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right)$ y $\left(\frac{1}{\omega}D - L\right)^{-1} b$:

$$\frac{1}{\omega}D = 2 * \begin{bmatrix} -3/10 & 0 & 0 & 0 \\ 0 & 1/10 & 0 & 0 \\ 0 & 0 & 1/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix} = \begin{bmatrix} -3/5 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 7/5 & 0 \\ 0 & 0 & 0 & 2/5 \end{bmatrix},$$

$$\frac{1}{\omega}D - L = \begin{bmatrix} -3/5 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 7/5 & 0 \\ 0 & 0 & 0 & 2/5 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1/10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -3/5 & 0 & 0 & 0 \\ 1/10 & 1/5 & 0 & 0 \\ 0 & 0 & 7/5 & 0 \\ 0 & 0 & 0 & 2/5 \end{bmatrix},$$

de donde

$$\left(\frac{1}{\omega}D - L\right)^{-1} = \begin{bmatrix} -5/3 & 0 & 0 & 0 \\ 5/6 & 5 & 0 & 0 \\ 0 & 0 & 5/7 & 0 \\ 0 & 0 & 0 & 5/2 \end{bmatrix}.$$

Además,

$$\begin{aligned} \frac{1-\omega}{\omega}D + U &= \begin{bmatrix} -3/10 & 0 & 0 & 0 \\ 0 & 1/10 & 0 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix} + \begin{bmatrix} 0 & -1/10 & 0 & 0 \\ 0 & 0 & 1/10 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} -3/10 & -1/10 & 0 & 0 \\ 0 & 1/10 & 1/10 & 0 \\ 0 & 0 & 7/10 & 0 \\ 0 & 0 & 0 & 2/10 \end{bmatrix}. \end{aligned}$$

Por lo que

$$\left(\frac{1}{\omega}D-L\right)^{-1}b=\begin{bmatrix}-5/3&0&0&0\\5/6&5&0&0\\0&0&5/7&0\\0&0&0&5/2\end{bmatrix}\begin{bmatrix}6\\2\\0\\3\end{bmatrix}=\begin{bmatrix}-10\\15\\0\\15/2\end{bmatrix},$$

y

$$\left(\frac{1}{\omega}D-L\right)^{-1}\left(\frac{1-\omega}{\omega}D+U\right)=\begin{bmatrix}-5/3&0&0&0\\5/6&5&0&0\\0&0&5/7&0\\0&0&0&5/2\end{bmatrix}\begin{bmatrix}-3/10&-1/10&0&0\\0&1/10&1/10&0\\0&0&7/10&0\\0&0&0&2/10\end{bmatrix}$$

$$=\begin{bmatrix}1/2&1/6&0&0\\-1/4&5/12&1/2&0\\0&0&1/2&0\\0&0&0&1/2\end{bmatrix}.$$

Así, las primeras tres iteraciones serán:

$$x^{(1)}=\begin{bmatrix}1/2&1/6&0&0\\-1/4&5/12&1/2&0\\0&0&1/2&0\\0&0&0&1/2\end{bmatrix}\begin{bmatrix}0\\0\\0\\0\end{bmatrix}+\begin{bmatrix}-10\\15\\0\\15/2\end{bmatrix}=\begin{bmatrix}-10\\15\\0\\15/2\end{bmatrix},$$

$$x^{(2)}=\begin{bmatrix}1/2&1/6&0&0\\-1/4&5/12&1/2&0\\0&0&1/2&0\\0&0&0&1/2\end{bmatrix}\begin{bmatrix}-10\\15\\0\\15/2\end{bmatrix}+\begin{bmatrix}-10\\15\\0\\15/2\end{bmatrix}=\begin{bmatrix}-75/6\\285/12\\0\\45/4\end{bmatrix}$$

y

$$x^{(3)}=\begin{bmatrix}1/2&1/6&0&0\\-1/4&5/12&1/2&0\\0&0&1/2&0\\0&0&0&1/2\end{bmatrix}\begin{bmatrix}-75/6\\285/12\\0\\45/4\end{bmatrix}+\begin{bmatrix}-10\\15\\0\\15/2\end{bmatrix}=\begin{bmatrix}885/72\\1345/48\\0\\105/8\end{bmatrix}.$$

2.7. Implementación computacional del método de relajación

El criterio de paro que se va a implementar es el mismo que se utilizó en la implementación de los métodos de Jacobi y Gauss-Seidel, que está definido en (2.5). La siguiente función está escrita en *OCTAVE*.

```
function Mrelajacion(A,b,x0,tol,w,itermax)
%
%%%%%%%%%%%%%%%%%%%%%% Datos de entrada %%%%%%%%%%%%%%%%%%%%%%
%
% A es una matriz cuadrada de tamaño n x n.
% b es un vector de tamaño n x 1.
% x0 es el punto de inicio.
% w es el parámetro de relajación. w debe ser distinto de cero.
% itermax es el número de interacciones máximas.
%
%%%%%%%%%%%%%%%%%%%%%% Datos de salida %%%%%%%%%%%%%%%%%%%%%%
%
% k el numero de iteracion.
% x_max es el vector en la iteracion k
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Universidad Juárez Autónoma de Tabasco.
% Fecha:6/noviembre/2018.
% Autores: Edwin E.,Justino A., Victoria O.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Descomposición de A = (1/w)D - L - ((1-w)/w)D - U,
%
clc
format long
x0 = x0(:); b = b(:);
D = diag(diag(A));
L = -tril(A,-1);
U = -triu(A,1);
M = (1/w)*D-L;
N = ((1-w)/w)*D + U
%%
```

```
%% Método de Relajación.
%%
x = x0;
for k = 1:itermax
k, p = N*x + b;
x_max = linsolve(M,p)
Error = norm(x_max-x, inf)/(norm(x_max, inf)+eps);
if Error < tol
break
end
x = x_max;
endfor
endfunction
```

Ejemplo 2.7.1. *Aproximar la solución de $Ax = b$ del ejemplo 2.1.1 con una tolerancia de 10^{-7} usando el método de relajación, y comparar los resultados con los obtenidos en el ejemplo 2.2.1 con el método de Jacobi y con el ejemplo 2.4.1 con el método de Gauss-Seidel.*

Solución. Sólo falta calcular la solución del sistema con el método de relajación. Para ello, aplicamos la función **Mrelajacion.m** en *OCTAVE*, cuya sintaxis es

$$\textbf{Mrelajacion}(A, b, x^{(0)}, tol, w, itermax),$$

donde $x^{(0)}$ es el punto de inicio del método, *tol* es la tolerancia para el criterio de paro e *itermax* es el número máximo de iteraciones permitidas en caso de que no hubiera convergencia.
Puesto que la matriz A de coeficientes del sistema es tridiagonal, se podría intentar aplicar el teorema 2.6.5 para calcular el parámetro óptimo de relajación, sin embargo, no todos los valores propios de $E_J = D^{-1}(L + U)$ son reales. Así que con ensayo y error, encontramos que con el parámetro $\omega = 0.9$, se obtiene buena velocidad de convergencia para el método de relajación.
En la tabla 2.6 presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un *itermax* de 50 y $\omega = 0.9$, y los comparamos con los resultados obtenidos con los métodos de Jacobi y Gauss-Seidel. Observamos que el método de relajación con $\omega = 0.9$, converge más rápido que los otros dos. En cambio, el método de Gauss-Seidel converge dos veces más rápido que el método de Jacobi, tal como lo dice el corolario 2.6.1.

Ejemplo 2.7.2. *Aproximar la solución del sistema $Ax=b$ donde*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ 0 \\ 3 \\ -6 \end{bmatrix}$$

con una tolerancia de 10^{-7}, usando los métodos de Jacobi, Gauss-Seidel y el método de relajación.

	Método de SOR ($\omega_b = 1.1010$)		Método de Gauss-Seidel		Método de Jacobi	
$x^{(0)}$	k	$x^{(k)}$	k	$x^{(k)}$	k	$x^{(k)}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	9	$\begin{bmatrix} -10.00000032256746 \\ 30.00000038149229 \\ 0.00000000000000 \\ 14.99999985000000 \end{bmatrix}$	16	$\begin{bmatrix} -9.999999303082808 \\ 29.999999303082809 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999303082806 \\ 30.000000696917191 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	9	$\begin{bmatrix} -10.00000016318799 \\ 30.00000039186901 \\ 9.999999999999965 \times 10^{-9} \\ 14.99999986000000 \end{bmatrix}$	16	$\begin{bmatrix} -9.999999319344209 \\ 29.999999319344212 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999326313377 \\ 30.000000696917191 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$	9	$\begin{bmatrix} -10.00000046979382 \\ 30.00000034124178 \\ -9.999999999999965 \times 10^{-9} \\ 14.99999985000000 \end{bmatrix}$	16	$\begin{bmatrix} -9.999999333282553 \\ 29.999999333282553 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$	31	$\begin{bmatrix} -9.999999303082806 \\ 30.000000766608913 \\ 0.000000000000000 \\ 15.000000000000000 \end{bmatrix}$

Tabla 2.6: Comparación del método de relajación ($\omega = 0.9$) con los métodos de Gauss-Seidel y Jacobi en la aproximación de la solución del sistema tridiagonal de $Ax = b$ del ejemplo 2.7.1, donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada. Observamos que el método de relajación converge más rápido que los otros dos métodos.

Solución. Notemos primero que la solución exacta del sistema es $x = (26, -63, 56, -17)^T$.
Antes de hacer la comparación, buscaremos el mejor valor de ω que haga que el método de SOR correspondiente converja más rápido a la solución. Elegimos como punto de inicio a $x^{(0)} = (0, 0, 0, 0)^T$. En la tabla 2.7 se muestran resultados para tres valores diferentes de ω, donde observamos que con $\omega = 1.7$, el método de SOR correspondiente, converge mucho más rápido a la solución de $Ax = b$.
También es interesante notar que los radios espectrales de las matrices de iteración de cada uno de los métodos son: $\rho(E_J) = 1.9268$, $\rho(E_\omega) = 0.98076$ y $\rho(E_{1.7}) = 0.89238$, donde el último es el radio espectral del método de SOR con $\omega = 1.7$. Es claro que el método de Jacobi no convergerá desde ningún punto de inicio, pues $\rho(E_J) = 1.9268 > 1$.
En la tabla 2.8, presentamos los resultados obtenidos desde tres puntos de inicio distintos, considerando un $itermax = 2000$. Notamos que el método de SOR con $\omega = 1.7$ converge desde cada uno de los puntos de inicio al igual que el método de Gauss-Seidel, pero el método de SOR converge mucho más rápido que el método de Gauss-Seidel. Notamos también que el método de Jacobi diverge desde los tres puntos de inicio como era de esperarse, de hecho los valores de $x^{(k)}$ se vuelven muy grandes, hasta desbordarse al infinito.
En el capítulo 3 presentaremos una aplicación de estos métodos iterativos en la solución numérica de la ecuación de Poisson en uno y dos dimensiones.

ω	k	$x^{(k)}$
0.8	920	$\begin{bmatrix} 25.9998066285275 \\ -62.9995142630053 \\ 55.9995832737748 \\ -16.9998787238106 \end{bmatrix}$
1.7	120	$\begin{bmatrix} 25.9999708707413 \\ -62.9999184251922 \\ 55.9999290645758 \\ -16.9999798771842 \end{bmatrix}$
1.9	367	$\begin{bmatrix} 25.9999431238028 \\ -62.9998490784306 \\ 55.9998736319151 \\ -16.9999654529226 \end{bmatrix}$

Tabla 2.7: Aproximaciones de la solución de $Ax = b$ del ejemplo 2.7.2 con el método de SOR con tres valores diferentes de ω. Donde k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada.

	Método de SOR ($\omega = 1.7$)		Método de Gauss-Seidel		Método de Jacobi	
$x^{(0)}$	k	$x^{(k)}$	k	$x^{(k)}$	k	$x^{(k)}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	120	$\begin{bmatrix} 25.9999708707413 \\ -62.9999184251922 \\ 55.9999290645758 \\ -16.9999798771842 \end{bmatrix}$	631	$\begin{bmatrix} 25.9998706340567 \\ -62.9996811433450 \\ 55.9997306462140 \\ -16.9999226261409 \end{bmatrix}$	1085	$\begin{bmatrix} 1.12966100626517 \times 10^{308} \\ 1.16540491851637 \times 10^{308} \\ 6.83561838291167 \times 10^{307} \\ 3.27664892383040 \times 10^{307} \end{bmatrix}$
$\begin{bmatrix} 0 \\ -1 \\ 2 \\ 3 \end{bmatrix}$	122	$\begin{bmatrix} 25.9999732161545 \\ -62.9999224650517 \\ 55.9999313639656 \\ -16.9999802847994 \end{bmatrix}$	633	$\begin{bmatrix} 25.9998703061674 \\ -62.9996803351748 \\ 55.9997299635132 \\ -16.9999224300300 \end{bmatrix}$	1080	$\begin{bmatrix} 1.31352100570909 \times 10^{308} \\ 1.35508248238907 \times 10^{308} \\ 7.94816169024966 \times 10^{307} \\ 3.80994578543065 \times 10^{307} \end{bmatrix}$
$\begin{bmatrix} -6 \\ -3 \\ -9 \\ -12 \end{bmatrix}$	130	$\begin{bmatrix} 25.9999929197113 \\ -62.9999978065058 \\ 56.0000065569229 \\ -17.0000035665683 \end{bmatrix}$	622	$\begin{bmatrix} 25.9998718906315 \\ -62.9996842405065 \\ 55.9997332625376 \\ -16.9999233776991 \end{bmatrix}$	1077	$\begin{bmatrix} 9.66141204877766 \times 10^{307} \\ 9.96711142458950 \times 10^{307} \\ 5.84615432764676 \times 10^{307} \\ 2.80235001622058 \times 10^{307} \end{bmatrix}$

Tabla 2.8: Comparación del método de SOR con los métodos de Gauss-Seidel y Jacobi en la aproximación de la solución de $Ax = b$ del ejemplo 2.7.2, desde tres puntos de inicio distintos. Donde $x^{(0)}$ es el punto de inicio, k es el mínimo número de iteraciones para lograr convergencia y $x^{(k)}$ es la aproximación alcanzada. Observamos que el método de SOR es casi cinco veces más rápido que el método de Gauss-Seidel, y el método de Jacobi de plano no converge.

Capítulo 3

Método de Thomas

Los métodos de discretización para problemas de valores en la frontera frecuentemente conducen a resolver sistemas de ecuaciones lineales con matrices con forma de banda, tridiagonales por bloques o *sparse*. Explotando la estructura de estas matrices se reduce dramáticamente el coste computacional de la factorización y de los algoritmos de sustitución para resolverlos [6]. Puesto que en la sección 4.2 del Capítulo 4, abordaremos la discretización del problema de valores en la frontera tipo Dirichlet para la ecuación de Poisson en una dimensión, que conduce a resolver sistemas de ecuaciones lineales con matrices tridiagonales; y en la sección 4.3 del mismo capítulo, abordaremos la discretización del problema de valores en la frontera también tipo Dirichlet para la ecuación de Poisson en dos dimensiones, que conduce a su vez a resolver sistemas de ecuaciones lineales con matrices tridiagonales por bloques (*sparse*); revisaremos en este capítulo uno de los métodos directos más populares, el método de Thomas[1], para resolver sistemas tridiagonales de ecuaciones lineales. A diferencia de los métodos iterativos revisados en los capítulos 1 y 2, el método de Thomas es un método directo para resolver sistemas tridiagonales de ecuaciones lineales. Las principales referencias son: *Numerical Mathematics* de Quarteroni, Sacco y Saleri (2007)[6] y *An Introduction to Numerical Analysis* de Süli y Mayers (2003)[10].

[1]Llewellyn Hilleth Thomas (1903–1992). Fue un físico británico y matemático aplicado. Es mejor conocido por sus contribuciones a la física atómica y molecular y a la física del estado sólido. En la matemática aplicada, elaboró un método eficiente para resolver sistemas tridiagonales de ecuaciones lineales (algoritmo de Thomas). Más información en *https://en.wikipedia.org/wiki/Llewellyn_ Thomas*.

3.1. Matrices tridiagonales

Definición 3.1.1. *Sea $n \geq 3$. Una matriz $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ se llama **tridiagonal** si solamente la diagonal principal y las dos diagonales adyacentes tienen elementos distintos de cero, es decir,*

$$a_{ij} = 0 \quad \text{si} \quad |i-j| > 1, \qquad i,j \in \{1,2,\ldots,n\}.$$

Sea A una matriz tridiagonal y no singular A, entonces A se escribe en la forma:

$$A = \begin{bmatrix} a_1 & c_1 & & 0 \\ b_2 & a_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ 0 & & b_n & a_n \end{bmatrix}. \tag{3.1}$$

En tal caso, las matrices L y U de la factorización LU de A son matrices bidiagonales de la forma:

$$L = \begin{bmatrix} 1 & & & 0 \\ \beta_2 & 1 & & \\ & \ddots & \ddots & \\ 0 & & \beta_n & 1 \end{bmatrix} \quad \text{y} \quad U = \begin{bmatrix} \alpha_1 & \gamma_1 & & 0 \\ & \alpha_2 & \ddots & \\ & & \ddots & \gamma_{n-1} \\ 0 & & & \alpha_n \end{bmatrix}. \tag{3.2}$$

Puesto que $A = LU$, se sigue que $\gamma_i = c_i$, $i = 1 : n-1$, y los coeficientes α_i y β_i se pueden calcular como:

$$\alpha_1 = a_1, \quad \beta_i = \frac{b_i}{\alpha_{i-1}}, \quad \alpha_i = a_i - \beta_i c_{i-1}, \qquad i = 2 : n. \tag{3.3}$$

Esto se conoce como el **algoritmo de Thomas** y puede considerarse como un caso particular de la factorización de Doolittle sin pivoteo [6]. Cuando no se está interesado en guardar los coeficientes de la matriz original, las entradas α_i y β_i pueden sobrescribirse en A.
El algoritmo de Thomas se puede aplicar para resolver un sistema lineal de ecuaciones $Ax = f$, donde A es una matriz tridiagonal no singular y $f \in \mathbb{R}^n$. Esto equivale a resolver dos sistemas bidiagonales $Ly = f$ y $Ux = y$.
Para el sistema $Ly = f$, obtenemos

$$y_1 = f_1, \quad y_i = f_i - \beta_i y_{i-1}, \quad i = 2 : n, \tag{3.4}$$

y para el sistema $Ux = y$, tenemos

$$x_n = \frac{y_n}{\alpha_n}, \quad x_i = \frac{y_i - c_i x_{i+1}}{\alpha_i}, \quad i = n-1 : 1. \tag{3.5}$$

El algoritmo requiere únicamente $8n-7$ flops: $3(n-1)$ flops para la factorización (3.3) y $5n-4$ flops para el proceso de sustitución (3.4) y (3.5) [6].

Existe una clase importante de problemas donde el pivoteo no es necesario, tal es el caso de las matrices simétricas y definidas positivas. Probaremos a continuación que también es el caso de las matrices tridiagonales y estrictamente diagonal dominantes.

Teorema 3.1.1 (Süli and Mayers (2003) [10]). *Sea $n \geq 3$ y $A \in \mathbb{R}^{n\times n}$ tridiagonal y estrictamente diagonal dominante. Entonces A es no singular y se factoriza, sin pivoteo, en la forma $A = LU$, donde $L \in \mathbb{R}^{n\times n}$ es triangular inferior con unos en la diagonal principal y $U \in \mathbb{R}^{n\times n}$ es triangular superior.*

Demostración. Supongamos que A es de la forma (3.1) y que la diagonal principal de U consta de $\alpha_1, \alpha_2, \ldots, \alpha_n$ como en (3.2). Primero mostremos por inducción sobre j que $|\alpha_j| > |c_j|$ para toda $j = 1 : n$.
Para $j = 1$, por ser A estrictamente diagonal dominante, se sigue de (3.3) que

$$|\alpha_1| = |a_1| > |c_1|.$$

Hipótesis de inducción: suponemos que para alguna $j \in \{2, 3, \ldots, n\}$, se satisface

$$|\alpha_j| > |c_j|.$$

Probemos que la desigualdad es cierta para $j + 1$. De (3.3) se sigue que

$$\begin{aligned} |\alpha_{j+1}| &= |a_{j+1} - \beta_{j+1}c_j| \\ &\geq ||a_{j+1}| - |\beta_{j+1}||c_j|| \\ &= \left| |a_{j+1}| - |b_{j+1}|\frac{|c_j|}{|\alpha_j|} \right|. \end{aligned} \tag{3.6}$$

Por otro lado, se sigue de la hipótesis de inducción que $|\alpha_j| > |c_j|$, de donde

$$\begin{aligned} \frac{|c_j|}{|\alpha_j|} &< 1 \\ -|b_{j+1}|\frac{|c_j|}{|\alpha_j|} &> -|b_{j+1}| \\ |a_{j+1}| - |b_{j+1}|\frac{|c_j|}{|\alpha_j|} &> |a_{j+1}| - |b_{j+1}|. \end{aligned} \tag{3.7}$$

Por ser A estrictamente diagonal dominante:

$$|a_{j+1}| - |b_{j+1}| > |c_{j+1}|. \tag{3.8}$$

Por lo que se sigue de (3.7) y (3.8) que

$$|a_{j+1}| - |b_{j+1}|\frac{|c_j|}{|\alpha_j|} > |c_{j+1}| \geq 0.$$

Concluimos ahora de (3.6) que

$$|\alpha_{j+1}| > |c_{j+1}|.$$

Por lo tanto, $|\alpha_j| > |c_j|$ para toda $j = 1 : n$.

En particular, $\alpha_j \neq 0$ para toda $j \in \{1, 2, \ldots, n\}$, por lo que la factorización $A = LU$ definida en (3.3) existe. Además,

$$\det(A) = \det(L)\det(U) = \det(U) = \alpha_1\alpha_2 \ldots \alpha_n \neq 0,$$

de modo que A es no singular.

Las igualdades (3.3) y las desigualdades $|\alpha_i| > |c_i|$, $i = 1 : n$, implican que

$$\begin{aligned} |\alpha_i| &\leq |a_i| + |\beta_i||c_{i-1}| \\ &= |a_i| + \frac{|b_i||c_{i-1}|}{|\alpha_{i-1}|} \\ &\leq |a_i| + |b_i|, \qquad \forall\, i = 1 : n. \end{aligned}$$

Así que los elemento α_i no pueden crecer demasiado, y por lo tanto, los errores de redondeo se mantienen bajo control sin necesidad de pivoteo. □

3.2. Estabilidad

Si A es una matriz no singular y tridiagonal, y $\widehat{L}$ y $\widehat{U}$ son los factores calculados de tal manera que

$$A + \delta A = \widehat{L}\widehat{U}.$$

Entonces

$$\|\delta A\| \leq (4u + 3u^2 + u^3)\|\widehat{L}\|\|\widehat{U}\|,$$

donde u es la unidad de redondeo de la aritmética de punto flotante en que se está trabajando (Quarteroni *et al.* (2007) [6]). En particular, si A es una matriz simétrica y definida positiva, entonces

$$\|\delta A\| \leq \frac{4u + 3u^2 + u^3}{1 - u}\|A\|,$$

lo cual implica la estabilidad del procedimiento de factorización en tales casos. Un resultado similar se mantiene incluso si A es de diagonal dominante.

3.3. Implementación computacional

Con respecto al caso tridiagonal, el algoritmo de Thomas se puede implementar de varias maneras. En particular, al implementarlo en computadoras donde las divisiones son más costosas que las multiplicaciones, es posible (y conveniente) diseñar una versión del algoritmo sin divisiones en (3.5), recurriendo a la siguiente forma de factorización [6]:

$$A = LDM^T = \begin{bmatrix} \gamma_1^{-1} & 0 & & \mathbf{0} \\ b_2 & \gamma_2^{-1} & \ddots & \\ & \ddots & \ddots & 0 \\ \mathbf{0} & & b_n & \gamma_n^{-1} \end{bmatrix} \begin{bmatrix} \gamma_1 & & & \mathbf{0} \\ & \gamma_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \gamma_n \end{bmatrix} \begin{bmatrix} \gamma_1^{-1} & c_1 & & \mathbf{0} \\ 0 & \gamma_2^{-1} & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ \mathbf{0} & & 0 & \gamma_n^{-1} \end{bmatrix}$$

Los coeficientes γ_i pueden calcularse recursivamente por las fórmulas

$$\gamma_i = (a_i - b_i \gamma_{i-1} c_{i-1})^{-1}, \quad i = 1 : n,$$

donde se han considerado $\gamma_0 = 0$, $b_1 = 0$ y $c_n = 0$. Los algoritmos de sustitución hacia adelante y hacia atrás, respectivamente, son, para el sistema $Ly = f$:

$$y_1 = \gamma_1 f_1, \quad y_i = (f_i - b_i y_{i-1}), \quad i = 2 : n, \tag{3.9}$$

y para el sistema $Ux = y$:

$$x_n = y_n, \quad x_i = y_i - \gamma_i c_i x_{i+1}, \quad i = n-1 : 1. \tag{3.10}$$

En la función siguiente, escrita en *OCTAVE*, mostramos una implementación del algoritmo de Thomas usando (3.9) y (3.10), sin divisiones. Los vectores de entrada a, b y c contienen los coeficientes de la matriz tridiagonal $\{a_i\}$, $\{b_i\}$ y $\{c_i\}$, respectivamente, mientras que el vector f contiene los componentes f_i del lado derecho f del sistema $Ax = f$.

```
function x = modthomas (a,b,c,f)
%
% function modthomas (a,b,c,f)
% Version modificada del algoritmo de Thomas
% x = modthomas (a,b,c,f) resuelve el sistema Ax=f,
% donde A es la matriz tridiagonal A=tridiag(b,a,c).
%%%%%%%%% Datos de entrada %%%%%%%%%%%%%%%%
%
% Considere el sistema Ax=f, donde A es tridiagonal.
% a, b y c son los vectores de entrada que contienen los coeficientes
% de la matriz tridiagonal A.
% f es el vecto que contiene las componentes f_i del lado derecho f.
```

```
%
%%%%%%%%% Datos de salida %%%%%%%%%%%%%%
%
% x es el vector que contiene la solucion del sistema Ax=f.
%
% Tomado de Quarteroni et al. (2007).
% Universidad Juárez Autónoma de Tabasco.
% Fecha:10/julio/2020.
% Edwin E., Justino A.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=length(a);
b=[0; b];
c=[c; 0];
gamma(1)=1/a(1);
for i=2:n
gamma(i)=1/(a(i)-b(i)*gamma(i-1)*c(i-1));
end
y(1)=gamma(1)*f (1);
for i =2:n
y(i)=gamma(i)*(f(i)-b(i)*y(i-1));
end
x(n,1)=y(n);
for i=n-1:-1:1
x(i,1)=y(i)-gamma(i)*c(i)*x(i+1,1);
end
endfunction
```

Veamos el siguiente ejemplo.

Ejemplo 3.3.1. *Aproximar la solución del sistema tridiagonal $Ax = b$, donde*

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} 2 \\ -3 \\ 8 \\ -7 \end{bmatrix},$$

con una tolerancia de 10^{-7}, usando los métodos de Thomas, Jacobi, Gauss-Seidel y el método de SOR.

Solución. Notemos primero que la solución exacta del sistema es $x = (1.6, 1.2, 3.8, -1.6)^T$. Por otro lado, como $\rho(E_J) = 0.809$, entonces aplicando el teorema 2.6.5, obtenemos que el

ω óptimo para la convergencia del método de SOR es $\omega_b = 1.26$. Haremos una comparación entre el método de Thomas, que es un método directo, con los métodos iterativos, midiendo el tiempo del CPU en segundos que requiere cada uno para obtener la solución del sistema tridiagonal. Los resultados se muestran en la tabla 3.1. Por tratarse de un sistema pequeño, el único método donde se registra un tiempo positivo del uso de CPU es el método de Jacobi. En este caso, el error que se reporta es el error relativo máximo, es decir,

$$\text{Error} = \max_{1 \leq i \leq 4} \frac{|x_i - \widehat{x}_i|}{|x_i|},$$

donde $\widehat{x} = (\widehat{x}_1, \widehat{x}_2, \widehat{x}_3, \widehat{x}_4)^T$ es la solución numérica. El punto de inicio para los métodos iterativos fue el $x^{(0)} = (0, 0, 0, 0)^T$. También observamos que el método de SOR empleó la mitad de iteraciones que empleó el método de Gauss-Seidel para alcanzar con convergencia con la tolerancia 10^{-7}. El método de Gauss-Seidel a su vez invirtió menos de la mitad de iteraciones que empleó el método de Jacobi. En las secciones 4.2 y 4.3 de Capítulo 4, presentaremos ejemplos más significativos.

Thomas		SOR			Gauss-Seidel			Jacobi		
TCPU	Error	TCPU	Itermax	Error	TCPU	Itermax	Error	TCPU	Itermax	Error
0	7.401×10^{-16}	0	16	6.266×10^{-8}	0	35	6.148×10^{-7}	0.031	77	1.353×10^{-7}

Tabla 3.1: Comparación del método de Thomas con los métodos iterativos, en la solución del sistema tridiagonal del ejemplo 3.3.1.

3.4. Sistemas de bloques

En esta sección se revisa la factorización LU de matrices particionadas en bloques, donde los bloque pueden ser de diferentes tamaños. El propósito es optimizar la ocupación de almacenamiento explotando adecuadamente la estructura de la matriz y reduciendo el coste computacional de la solución del sistema. Aclaramos que la sección está basada en el libro de Quarteroni *et al.* (2007) [6].

3.4.1. Factorización LU por bloques

Sea $A \in \mathbb{R}^{n \times n}$ la siguiente matriz particionada por bloques

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

donde $A_{11} \in \mathbb{R}^{r\times r}$ es una matriz cuadrada no singular cuya factorización $L_{11}D_1R_{11}$ se conoce, mientras que $A_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$. En tal caso, es posible factorizar A usando sólo la factorización LU del bloque A_{11}. De hecho, se tiene que

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{n-r} \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & \Delta_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & I_{n-r} \end{bmatrix},$$

donde

$$\begin{aligned} L_{21} &= A_{21}R_{11}^{-1}D_1^{-1}, \quad R_{12} = D_1^{-1}L_{11}^{-1}A_{12}, \\ \Delta_2 &= A_{22} - L_{21}D_1R_{12}. \end{aligned}$$

Si fuera necesario, el procedimiento de reducción puede repetirse en la matriz Δ_2, obteniendo así una versión en bloque de la factorización LU.
Si A_{11} fuera un escalar, el enfoque anterior se reduce a la factorización estándar de una matriz dada. La aplicación iterativa de este método produce una forma alternativa de realizar la eliminación de Gauss. Se tiene el siguiente resultado para matrices de bloques.

Teorema 3.4.1 (Quarteroni *et al.* (2007) [6]). *Sea $A \in \mathbb{R}^{n\times n}$ particionada en $m \times m$ bloques A_{ij} con $i, j = 1 : m$. Entonces A admite una única factorización por bloque LU (con unos en la diagonal principal de L) si y sólo si los $m - 1$ bloques principales dominantes menores de A son distintos de cero.*

Dado que la factorización por bloque es una formulación equivalente de la factorización LU estándar de A, el análisis de estabilidad llevado a cabo para éste último también es válido para la versión en bloque.

3.4.2. Inversa de una matriz particionada por bloques

El inverso de una matriz de bloques se puede construir utilizando la factorización LU presentada en la sección anterior. Una aplicación notable es cuando A es una matriz de bloques de la forma

$$A = C + UBV,$$

donde C es una matriz de bloques que es "fácil" de invertir (por ejemplo, cuando C está dada por los bloques diagonales de A), mientras que U, B y V toman en cuenta las conexiones entre los bloques diagonales. En tal caso, A se puede invertir utilizando la fórmula de Sherman-Morrison o Woodbury:

$$A^{-1} = (C + UBV)^{-1} = C^{-1} - C^{-1}U(I + BVC^{-1}U)^{-1}BVC^{-1},$$

bajo el supuesto que C e $I + BVC^{-1}U$ son matrices no singulares. Esta fórmula tiene varias aplicaciones prácticas y teóricas, y es particularmente efectiva si las conexiones entre bloques son de relevancia modesta.

3.4.3. Sistemas tridiagonales por bloques

Consideramos el sistema tridiagonal por bloques de la forma

$$A_n\mathbf{x} = \begin{bmatrix} A_{11} & A_{12} & & \mathbf{0} \\ A_{21} & A_{22} & \ddots & \\ & \ddots & \ddots & A_{n-1,n} \\ \mathbf{0} & & A_{n,n-1} & A_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}, \tag{3.11}$$

donde A_{ij} son matrices de orden $n_i \times n_j$ y $\mathbf{x}_i$ y $\mathbf{b}_i$ son vectores columnas de tamaño n_i, para $i, j = 1 : n$. Suponemos que los bloques diagonales son cuadrados, aunque no necesariamente del mismo tamaño. Para $k = 1 : n$, tenemos

$$A_k = \begin{bmatrix} I_{n_1} & & & \mathbf{0} \\ L_1 & I_{n_2} & & \\ & \ddots & \ddots & \\ \mathbf{0} & & L_{k-1} & I_{n_k} \end{bmatrix} \begin{bmatrix} U_1 & A_{12} & & \mathbf{0} \\ & U_2 & \ddots & \\ & & \ddots & A_{k-1,k} \\ \mathbf{0} & & & U_k \end{bmatrix}.$$

Tomando $k = n$, y comparando los bloques correspondientes de la matriz anterior con A_n, resulta que $U_1 = A_{11}$, mientras que los bloques restantes se pueden obtener resolviendo sucesivamente, para $i = 2 : n$, los sistemas $L_{i-1}U_{i-1} = A_{i,i-1}$ para las columnas de L y calculando $U_i = A_{ii} - L_{i-1}A_{i-1,i}$.

Este procedimiento está bien definido solo si todas las matrices U_i son no singulares, que es el caso si, por ejemplo, las matrices $A_1, A_2, \ldots, A_n$ son no singulares. Como alternativa, se podría recurrir a métodos de factorización para matrices con bandas, incluso si esto requiere el almacenamiento de una gran cantidad de entradas cero (a menos que se realice un reordenamiento adecuado de las filas de la matriz).

Un ejemplo notable es cuando la matriz es tridiagonal por bloques y simétrica, con bloques simétricos y definidos positivos. En tal caso (3.11) toma la forma:

$$\begin{bmatrix} A_{11} & A_{21}^T & & \mathbf{0} \\ A_{21} & A_{22} & \ddots & \\ & \ddots & \ddots & A_{n,n-1}^T \\ \mathbf{0} & & A_{n,n-1} & A_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}. \tag{3.12}$$

Aquí consideramos una extensión, al caso de bloques, el algoritmo de Thomas, que tiene como objetivo transformar A en una matriz bidiagonal de bloques. Para este propósito, primero tenemos que eliminar el bloque correspondiente a la matriz A_{21}. Suponemos que la factorización de Cholesky de A_{11} es posible y denotemos por H_{11} el factor de Cholesky, es decir, $A_{11} = H_{11}^T H_{11}$. Si multiplicamos la primera fila del sistema de bloques (3.12) por H_{11}^{-T}, encontramos que

$$H_{11}\mathbf{x}_1 + H_{11}^{-T}A_{21}^T\mathbf{x}_2 = H_{11}^{-T}\mathbf{b}_1.$$

Poniendo $H_{21} = H_{11}^{-T} A_{21}^T$ y $\mathbf{c}_1 = H_{11}^{-T} \mathbf{b}_1$, se sigue que $A_{21} = H_{21}^T H_{11}$, por lo que las dos primeras filas del sistema (3.12) serán

$$\begin{aligned} H_{11}\mathbf{x}_1 + H_{21}\mathbf{x}_2 &= \mathbf{c}_1, \\ H_{21}^T H_{11}\mathbf{x}_1 + A_{22}\mathbf{x}_2 + A_{32}^T\mathbf{x}_3 &= \mathbf{b}_2. \end{aligned}$$

Como una consecuencia, multiplicando la primera fila por H_{21}^T y restándola de la segunda fila, se elimina la variable desconocida $\mathbf{x}_1$ y se obtiene la siguiente ecuación equivalente:

$$A_{22}^{(1)}\mathbf{x}_2 + A_{32}^T\mathbf{x}_3 = \mathbf{b}_2 - H_{21}^T\mathbf{c}_1,$$

con $A_{22}^{(1)} = A_{22} - H_{21}^T H_{21}$. En este punto, se lleva a cabo la factorización de $A_{22}^{(1)}$ y la variable desconocida x_2 se elimina de la tercera fila del sistema, y se repite el proceso para las filas restantes del sistema. Al final del procedimiento, que requiere resolver $(n-1)\sum_{j=1}^{n-1} n_j$ sistemas lineales para calcular las matrices $H_{i+1,i}$, $i = 1 : n-1$, se obtiene el siguiente sistema bidiagonal por bloques:

$$\begin{bmatrix} H_{11} & H_{21} & & \mathbf{0} \\ & H_{22} & \ddots & \\ & & \ddots & H_{n,n-1} \\ \mathbf{0} & & & H_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_n \end{bmatrix},$$

que se puede resolver con un método de sustitución regresiva (por bloques). Si todos los bloques tienen el mismo tamaño p, entonces el número de multiplicaciones que requiere el algoritmo es de aproximadamente $\frac{7}{6}(n-1)p^3$ flops (suponiendo que p y n son muy grandes).

3.5. Matrices *sparse*

En esta sección abordamos brevemente la solución numérica de sistemas *sparse* lineales, es decir, sistemas donde la matriz $A \in \mathbb{R}^{n\times n}$ tiene un número de entradas distintas de cero de orden $O(n)$ (y no $O(n^2)$). Llamamos un *patrón* de matriz dispersa al conjunto de sus coeficientes distintos de cero.

Las matrices de bandas con bandas suficientemente pequeñas son matrices *sparse*. Obviamente, para una matriz *sparse*, la estructura de la matriz en sí misma es redundante y puede ser sustituido convenientemente por una estructura tipo vector por medio de *técnicas de compactación de matrices*.

Por conveniencia, asociamos con una matriz *sparse* A un *grafo orientado* $G(A)$. Un grafo es un par $(\mathcal{V}, \mathcal{X})$ donde $\mathcal{V}$ es un conjunto de p puntos y $\mathcal{X}$ es un conjunto de q pares ordenados de elementos de $\mathcal{V}$ que están unidos por una línea. Los elementos de $\mathcal{V}$ se llaman *vértices* del grafo, mientras que las líneas de conexión se llaman *trayectorias* o *caminos* del grafo.

El grafo $G(A)$ asociado con una matriz $A \in \mathbb{R}^{m\times n}$ puede construirse identificando los vértices con el conjunto de los índices del 1 al máximo entre m y n, y suponiendo que existe un camino que conecta dos vértices i y j si $a_{ij} \neq 0$ y se dirige de i a j, para $i = 1 : m$ y $j = 1 : n$. Para una entrada diagonal $a_{ii} \neq 0$, la camino que une el vértice i consigo mismo se denomina *bucle*. Como una orientación está asociada con cada lado, el grafo se llama orientado (o dirigido finito). Como ejemplo, la figura 3.1 muestra el patrón de una matriz 12×12 simétrica y *sparse*, junto con su grafo asociado.
Como se observó anteriormente, durante el procedimiento de factorización, se pueden generar entradas distintas de cero en posiciones de memoria que corresponden a entradas cero en la matriz inicial. Esta acción se conoce como *relleno*. La figura 3.2 muestra el efecto del relleno sobre la matriz *sparse* cuyo patrón se muestra en la figura 3.1. Dado que el uso del *pivoteo* en el proceso de factorización complica aún más las cosas, solo consideraremos el caso de matrices simétricas y definidas positivas que no requieren necesariamente el pivoteo.

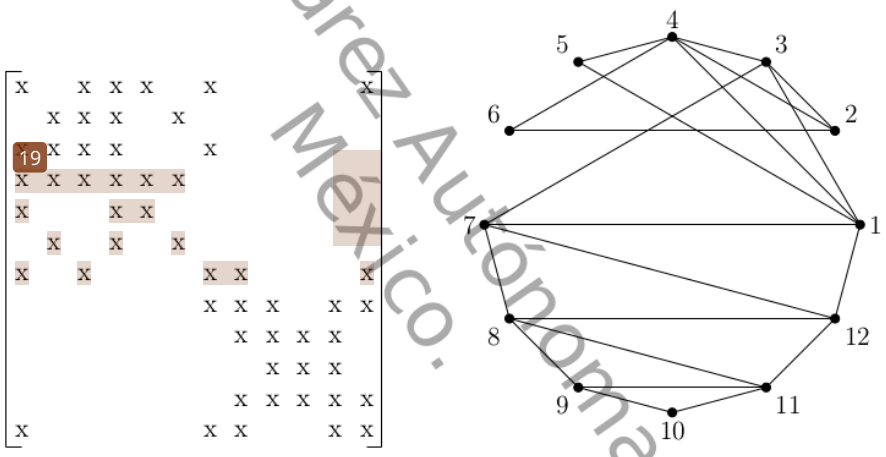


Figura 3.1: *Patrón* de una matriz *sparse* y simétrica (izquierda) y de su grafo asociado (derecha). Por claridad, los bucles no se han dibujado; además, dado que la matriz es simétrica, sólo se ha reportado uno de los dos lados asociados con cada $a_{ij} \neq 0$, [6].

Un primer resultado notable se refiere a la cantidad de relleno.
Sea

$$m_i(A) = i - \min\{j < i : a_{ij} \neq 0\}$$

y denotemos por $\mathcal{E}(A)$ el *casco convexo* (*convex hull* en inglés) de A, definido por

$$\mathcal{E}(A) = \{(i, j) : 0 < i - j \leq m_i(A)\}.$$

Para una matriz simétrica positiva definida, tenemos

$$\mathcal{E}(A) = \mathcal{E}(H + H^T), \tag{3.13}$$

donde H es el factor de Cholesky, de modo que el relleno está confinado dentro del casco convexo de A, como se observa en la figura 3.2. Además, si denotamos por $l_k(A)$ el número de filas activas en el $k-$ésimo paso de la factorización (es decir, el número de filas de A con $i > k$ y $a_{ik} \neq 0$), el coste computacional del proceso de factorización es

$$\frac{1}{2}\sum_{k=1}^{n} l_k(A)(l_k(A)+3) \quad \text{flops}, \tag{3.14}$$

habiendo contabilizado todas las entradas distintas de cero del casco convexo. El confinamiento del relleno dentro de $\mathcal{E}(A)$ asegura que la factorización LU de A se pueda almacenar sin áreas de memoria adicionales simplemente almacenando todas las entradas de $\mathcal{E}(A)$ (incluyendo los elementos nulos). Sin embargo, dicho procedimiento aún podría ser altamente ineficiente debido a la gran cantidad de entradas cero en el casco.

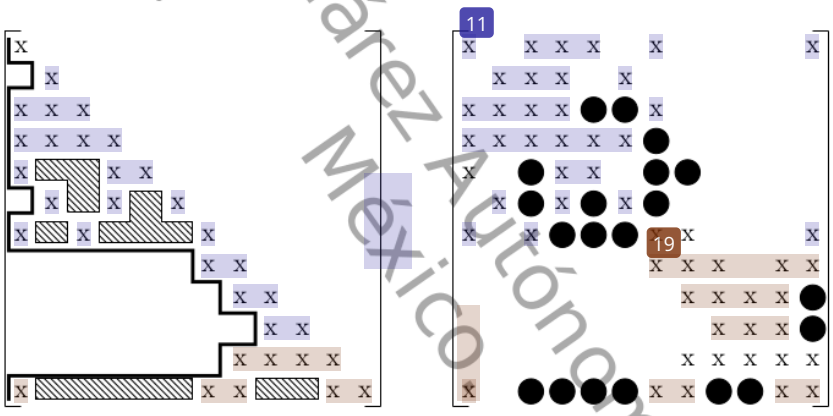


Figura 3.2: Las regiones sombreadas en la figura de la izquierda muestran las áreas de la matriz que pueden ser afectados por el relleno, para la matriz considerada en la figura 3.1. Las líneas continuas o sólidas denotan la frontera de $\mathcal{E}(A)$. La figura de la derecha muestra los factores que han sido actualmente calculados. Los puntos negros denotan los elementos de A que originalmente eran ceros, [6].

Por otro lado, de (3.13) se deduce que la reducción en el casco convexo refleja una reducción del relleno y, a su vez, debido a (3.14), del número de operaciones necesarias para realizar la factorización. Por esta razón, se han ideado varias estrategias para reordenar el grafo de una matriz, entre ellos, el método de Cuthill-McKee [6].
Otra alternativa consiste en descomponer la matriz en submatrices *sparse*, con el objetivo de reducir el problema original a la solución de subproblemas de tamaño reducido, donde las matrices se pueden almacenar en formato completo. Este enfoque conduce a métodos de descomposición de submatrices. Más detalles en [6].

Capítulo 4

Solución numérica de la ecuación de Poisson en una y dos dimensiones

Como ejemplo de aplicación de los métodos iterativos para resolver sistemas de ecuaciones lineales algebraicas revisados en los capítulos 1 y 2, así como la aplicación del método de Thomas revisado en el Capítulo 3, presentaremos en este capítulo, la solución numérica de la ecuación de Poisson en una y dos dimensiones espaciales con condiciones de frontera tipo Dirichlet. La discretización de este problema la haremos aplicando el método de diferencias finitas, que conducirá de manera natural a un sistema tridiagonal en el caso de una dimensión, y a un sistema tridiagonal por bloques en el caso de dos dimensiones. Las principales referencias son: *Applied Numerical Lineal Algebra* de Demmel (1997)[2], *Ecuaciones de la Física Matemática* de Tijonov y Samarsky (1980)[7], *Introduction to Numerical Analysis* de Stoer y Bulirsch (1993)[9], *Iterative Solution of Large Linear Systems* de Young (1971)[12] y *The eigenproblem of a tridiagonal 2-Toeplitz matrix* de Gover (1994)[3].

4.1. Ecuación de Poisson

Al estudiar procesos estacionarios de distinta naturaleza física, por ejemplo, oscilaciones, conducción del calor, difusión, entre otros, se obtienen, por lo general, ecuaciones de tipo elíptico. La ecuación más frecuente de este tipo es la ecuación de Laplace[1] [7]:

$$\Delta u = 0, \tag{4.1}$$

donde

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

es el operador de Laplace.
La función u se llama armónica en la región $\Omega \subset \mathbb{R}^N$, $N = 2, 3$, si es continua en esta región junto con sus derivadas parciales de hasta segundo orden y satisface la ecuación de Laplace (4.1).
Algunos ejemplos de funciones armónicas son:

a) $u(x, y) = ax + by + c, \quad (x, y) \in \Omega = \mathbb{R}^2$, a, b, c constantes reales.

b) $u(x, y, z) = x^2 - y^2 + 2z, \quad (x, y, z) \in \Omega = \mathbb{R}^3$.

c) $u(x, y) = \ln \sqrt{x^2 + y^2}, \quad (x, y) \in \Omega = \mathbb{R}^2 \setminus \{(0, 0)\}$.

d) $u(x, y) = e^x \operatorname{sen} y, \quad (x, y) \in \Omega = \mathbb{R}^2$.

e) $u(x, y, z) = -\ln\left(\sqrt{x^2 + y^2 + z^2} + z\right), \quad (x, y, z) \in \Omega = \mathbb{R}^2 \setminus \{(0, 0, z), z \leq 0\}$.

En el estudio de las propiedades de las funciones armónicas, se desarrollaron diferentes métodos matemáticos, que resultaron fructíferos también al aplicarlos a las ecuaciones de tipos hiperbólicos y parabólicos.

4.1.1. Propagación del calor en el espacio

El proceso de propagación del calor en el espacio se puede caracterizar por la temperatura

$$u(x, y, z, t)$$

que es función de x, y, z y t.
Si la temperatura no es constante, surgen flujos térmicos, dirigidos de los lugares de mayor temperatura hasta los de menor temperatura.

[1] Pierre-Simon Laplace (1749–1827) astrónomo, físico y matemático francés.

La ley de Fourier[2] establece que el vector de densidad del flujo térmico $\mathbf{W}$ tiene la forma

$$\mathbf{W} = -k \operatorname{grad} u,$$

donde k se conoce como el coeficiente de conductividad térmica. El coeficiente k depende del medio, por ejemplo, cuando el medio es isotrópico, k es un escalar.
La ecuación diferencial de la conducción del calor en un medio isotrópico (Tijonov y Samarsky 1980 [7]) está dada por

$$c\rho\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right) + F,$$

donde c es el calor específico, ρ es la densidad del medio y F es la fuente del calor en el medio.
Si el medio es homogéneo, entonces k, c y ρ se pueden considerar constantes. En este caso, la ecuación de calor toma la forma:

$$\frac{\partial u}{\partial t} = a^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f \tag{4.2}$$

o bien,

$$u_t = a^2\Delta u + f,$$

donde $a^2 = k/c\rho$ es el coeficiente de conductividad térmica de la temperatura y $f = F/c\rho$.
La ecuación de calor es el prototipo de una ecuación de tipo parabólico. Si el proceso es estacionario, se establece una distribución de temperatura $u(x, y, z)$ que no varía con el tiempo, por lo que (4.2) se reduce a la forma

$$\Delta u = -f, \tag{4.3}$$

donde ahora $f = F/k$. Esta ecuación se conoce como la ecuación de Poisson[3].
Cuando $f = 0$, es decir, cuando no hay fuentes del calor, se tiene la ecuación de Laplace:

$$\Delta u = 0.$$

[2]Jean-Baptiste Joseph Fourier (1768–1830) matemático y físico francés.
[3]Siméon Denis Poisson (1781–1840) físico y matemático francés.

4.1.2. Planteamiento de los problemas de contorno

Consideramos ahora un cierto volumen Ω, delimitado por una superficie $\partial\Omega$. El problema sobre la distribución estacionaria de la temperatura $u(x, y, z)$ dentro del cuerpo Ω se formula como sigue.

Problema de contorno. Encontrar la función $u(x, y, z)$ que satisfaga dentro de Ω a la ecuación de Poisson

$$\Delta u = -f(x, y, z), \quad (x, y, z) \in \operatorname{int}\Omega,$$

y a la condición de frontera[4], que puede ser tomada en una de las formas siguientes:

a) $u = f_1$ en $\partial\Omega$, primer problema de contorno o problema de Dirichlet,

b) $\dfrac{\partial u}{\partial n} = f_2$ en $\partial\Omega$, segundo problema de contorno o problema de Neumann,

c) $\dfrac{\partial u}{\partial n} + h(u - f_3) = 0$ en $\partial\Omega$, tercer problema de contorno o problema de Robin,

donde f_1, f_2, f_3 y h son funciones conocidas, y $\dfrac{\partial u}{\partial n}$ es la derivada normal en la dirección del vector normal exterior n a la superficie $\partial\Omega$, definida por

$$\frac{\partial u}{\partial n}(x, y, z) = \nabla u(x, y, z) \cdot n(x, y, z), \quad (x, y, z) \in \partial\Omega.$$

En un contexto físico, (4.3) describe el estado estable de distribución de temperatura de un objeto ocupando en la región Ω. La solución $u(x, y, z)$ representa la temperatura estable de la región Ω con fuentes de calor y cuencas dadas por $f(x, y, z)$. La condición de Dirichlet f_1 representa una temperatura fija en la frontera $\partial\Omega$, mientras que la condición de Newmann f_2 representa el flujo de calor en $\partial\Omega$. La condición de Newmann con $f_2 = 0$, representa una frontera perfectamente aislada.

[4]Dirichlet (1805–1859) matemático alemán. Neumann (1832–1925) matemático alemán. Robin (1855–1897) matemático francés.

4.2. Ecuación de Poisson en una dimensión

Un caso particular del problema de contorno tipo Dirichlet para la ecuación de Poisson en una dimensión es: dada $f : [0,1] \to \mathbb{R}$ continua, encontrar $u : [0,1] \to \mathbb{R}$ tal que

$$\left.\begin{array}{rcl} -u_{xx} & = & f(x)\,, \quad 0 < x < 1 \\ u(0) & = & 0 \\ u(1) & = & 0. \end{array}\right\} \tag{4.4}$$

Una estrategia para determinar aproximadamente la función incógnita u, consiste en discretizar el problema para calcular una solución aproximada en $N+2$ puntos x_i uniformemente espaciados entre 0 y 1, como sigue:

$$x_i = ih\,, \quad h = \frac{1}{N+1}\,, \quad i = 0 : N+1.$$

Recordando que la fórmula de diferencia central que aproxima la segunda derivada de una función $g(x)$ está dada por

$$g''(x) = \frac{g(x+h) - 2g(x) + g(x-h)}{h^2} + O(h^2)\,, \tag{4.5}$$

y aplicándolo a la función u, resulta para el operador diferencial que

$$-u_{xx} = -\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2)\,. \tag{4.6}$$

Si se abrevia

$$u_i := u(x_i)\,, \quad i = 0 : N+1,$$

resulta que el operador diferencial $-u_{xx}$ expresada en (4.6) puede ser reemplazado para todo x_i por el operador en diferencias

$$\frac{2u_i - u_{i-1} - u_{i+1}}{h^2} + \varepsilon_i\,, \quad i = 1 : N, \tag{4.7}$$

con un error ε_i, que se le llama **error de truncamiento** en cada punto x_i. Puesto que las condiciones de frontera (4.4) imponen que $u_0 = u_{N+1} = 0$, se sigue de (4.4) y (4.7) que los valores desconocidos u_i, $i = 1 : N$, deben satisfacer un sistema de ecuaciones lineales algebraicas de la forma

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i - h^2 \varepsilon_i\,, \quad i = 1 : N, \tag{4.8}$$

donde $f_i = f(x_i)$, $i = 1 : N$.

Es claro que los errores ε_i dependen del tamaño del paso h de la malla. Si la solución exacta u es suficientemente diferenciable, entonces $\varepsilon_i = O(h^2\|d^4u/dx^4\|_\infty)$ [2]. La escritura matricial del sistema (4.8) es

$$T_N u = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} - h^2 \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix} = h^2 f - h^2 \varepsilon, \qquad (4.9)$$

donde $u = (u_1, u_2, \ldots, u_N)^T$, $f = (f_1, f_2, \ldots, f_N)^T$ y $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N)^T$. Si se omite el error de truncamiento ε en (4.9), resulta el sistema tridiagonal lineal

$$T_N z = h^2 f, \qquad (4.10)$$

que cuando h es suficientemente pequeño, se espera que cada componente z_i, $i = 1 : N$, de la solución de (4.10) sea aproximadamente igual a u_i, $i = 1 : N$. En cada punto x_i le corresponde la componente z_i de la solución de (4.10). Es claro que el sistema tridiagonal lineal (4.10) se puede resolver con métodos directos como el método de Thomas (véase Capítulo 3) o con métodos iterativos vistos en el Capítulo 2.
La matriz de coeficientes T_N es de tamaño $N \times N$, simétrica, tridiagonal, diagonal dominante y normal (es decir, $T_N^T T_N = T_N T_N^T$). Sus valores y vectores propios se dan en el siguiente lema:

Lema 4.2.1. *Los valores propios de T_N son*

$$\lambda_i = 2\left(1 - \cos\frac{i\pi}{N+1}\right), \quad i = 1 : N,$$

y los vectores propios normalizados con respecto a la norma $\|\cdot\|_2$ son

$$v_i = \sqrt{\frac{2}{N+1}} \begin{bmatrix} \operatorname{sen}\dfrac{i\pi}{N+1} \\ \operatorname{sen}\dfrac{2i\pi}{N+1} \\ \vdots \\ \operatorname{sen}\dfrac{Ni\pi}{N+1} \end{bmatrix}, \quad i = 1 : N.$$

Demostración. Véase en el artículo de Gover (1994) [3]. □

De acuerdo con el teorema de descomposición real de Schur, véase por ejemplo Young (1971) [12] o Stoer and Bulirsch (1993) [9], la matriz $Q = [v_1, v_2, \ldots, v_N]$ es ortogonal y si $\Lambda =$

diag$\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$, entonces $T_N = Q\Lambda Q^T$. En este caso, el valor propio más grande de T_N es

$$\lambda_N = 2\left(1 - \cos\pi\frac{N}{N+1}\right) \cong 4,$$

y el valor propio más pequeño es λ_1, donde para cada $i = 1 : N-1$,

$$\lambda_i = 2\left(1 - \cos\frac{i\pi}{N+1}\right) \cong 2\left(1 - \left(1 - \frac{i^2\pi^2}{2(N+1)^2}\right)\right) = \left(\frac{i\pi}{N+1}\right)^2.$$

Así que T_N es definida positiva. Además, el hecho de que T_N sea normal, se garantiza que

$$\|T_N\|_2 = \text{máx}\{\lambda_1, \lambda_2, \ldots, \lambda_N\} \cong 4,$$

y de manera similar que

$$\|T_N^{-1}\|_2 = \text{máx}\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_N^{-1}\} = \lambda_1^{-1} \cong \left(\frac{N+1}{\pi}\right)^2.$$

Por lo que el número de condición de T_N (véase Demmel (1997) [2]) es

$$\kappa_2(T_N) = \|T_N\|_2\|T_N^{-1}\|_2 = \frac{\lambda_N}{\lambda_1} \cong \frac{4(N+1)^2}{\pi^2}$$

para N grande.
Con esto se puede acotar el error de truncamiento ε, es decir, la diferencia entre la solución aproximada z de (4.10) con la solución exacta u del problema de contorno (4.4). Para ello, restando (4.10) de (4.9) se tiene

$$u - z = -h^2 T_N^{-1}\varepsilon\,,$$

de donde

$$\begin{aligned}\|u - z\|_2 &\leq h^2\|T_N^{-1}\varepsilon\|_2 \leq h^2\|T_N^{-1}\|_2\|\varepsilon\|_2 \cong h^2\frac{(N+1)^2}{\pi^2}\|\varepsilon\|_2 \\ &= \frac{1}{\pi^2}\|\varepsilon\|_2 = O(\|\varepsilon\|_2) = O\left(h^2\left\|\frac{d^4u}{dx^4}\right\|_\infty\right).\end{aligned}$$

A continuación presentamos un ejemplo de la solución numérica de la ecuación de Poisson.

Ejemplo 4.2.1. *Encontrar la solución u del problema de contorno tipo Dirichlet de la ecuación de Poisson en una dimensión:*

$$\left.\begin{aligned} u_{xx} &= 5\,\text{sen}(7x), \quad -\pi < x < \pi \\ u(-\pi) &= 0 \\ u(\pi) &= 0. \end{aligned}\right\} \tag{4.11}$$

Solución. Primero encontramos la solución exacta u integrando como sigue:

$$\int u_{xx}dx = \int 5\,\text{sen}(7x)dx \quad \Longrightarrow \quad u_x(x) = -\frac{5}{7}\cos(7x) + C_1$$

$$\int u_x dx = \int (-\frac{5}{7}\cos(7x) + C_1)dx \quad \Longrightarrow \quad u(x) = -\frac{5}{49}\,\text{sen}(7x) + C_1 x + C_2.$$

Aplicando las condiciones de frontera, obtenemos

$$u(-\pi) = -\tfrac{5}{49}\,\text{sen}(-7\pi) - \pi C_1 + C_2 = 0$$
$$-\pi C_1 + C_2 = 0 \quad \Longrightarrow \quad C_2 = \pi C_1.$$

Por otro lado,

$$u(\pi) = -\tfrac{5}{49}\,\text{sen}(7\pi) + \pi C_1 + C_2 = 0$$
$$\pi C_1 + C_2 = \pi C_1 + \pi C_1 = 2\pi C_1 = 0$$
$$C_1 = 0.$$

Así que $C_1 = 0$ y $C_2 = 0$. Por lo tanto, la solución exacta es

$$u(x) = -\frac{5}{49}\,\text{sen}(7x), \quad -\pi \le x \le \pi. \tag{4.12}$$

Ahora aproximaremos valores de u en N puntos uniformemente espaciados x_i, $-\pi < x_i < \pi$, aplicando la técnica numérica anterior. Como aplicaremos los métodos iterativos de Jacobi, Gauss-Seidel y relajación, para resolver el sistema tridiagonal (4.10), necesitaremos elegir el ω óptimo que garantice una mayor velocidad de convergencia para el método de relajación. Para elegirlo, es posible aplicar el teorema 2.6.5 del Capítulo 2, ya que numéricamente se verifica que los valores propios de $E_J = D^{-1}(L+U)$ son reales y $\rho(E_J) < 1$. Los valores óptimos obtenidos de ω, así como los radios espectrales de las matrices de iteración de los métodos iterativos, se presentan en la tabla 4.1.

N	$\rho(E_J)$	$\rho(E_{GS})$	ω_b	$\rho(E_{\omega_b})$
50	0.99810	0.99621	1.8840	0.8840
75	0.99915	0.99829	1.9206	0.9206
100	0.99952	0.99903	1.9397	0.9397
300	0.99995	0.99989	1.9793	0.9793

Tabla 4.1: Valores óptimos ω_b para la convergencia del método de SOR. En este caso, se aplicó $\rho(E_J) = \max(\text{abs}(\text{eig}(E_J)))$ de OCTAVE.

En la tabla 4.2 se presenta una comparación entre los métodos iterativos de SOR óptimo, Gauss-Seidel y Jacobi, en la solución numérica del sistema tridiagonal (4.10). Se tomó como

punto de inicio a $x^{(0)} = (1, 1, \ldots, 1)^T \in \mathbb{R}^N$ en cada uno de ellos. Notamos que el método de SOR resultó más eficiente, al requerir menor cantidad de iteraciones (Iter) para alcanzar convergencia a la tolerancia 10^{-5}. El error que se reporta es el error absoluto máximo, es decir,

$$\text{Error} = \max_{1 \leq i \leq N} |u(x_i) - z_i|.$$

En la tabla 4.3 se reporta una comparación entre los métodos iterativos y el método de Thomas que es un método directo. En este caso, se hace una comparación del tiempo de CPU, en segundos, que requiere cada método para obtener la solución del sistema tridiagonal (4.10). Notamos que el método de Thomas resultó mucho más eficiente que los métodos iterativos.

En las figuras 4.1 al 4.4, se reportan las gráficas de la solución exacta $u(x) = -\frac{5}{49}\,\text{sen}(7x)$ pintada de azul, del problema de contorno (4.11), con la solución numérica $z = (z_1, z_2, \ldots, z_N)^T$ pintada de rojo y obtenida al resolver el sistema tridiagonal (4.10) con una tolerancia de 10^{-5}. Notamos que a medida que se incrementa el número de puntos uniformemente espaciados ($N = 50, 75, 100, 300$), la solución numérica se aproxima cada vez más a la solución exacta.

	SOR óptimo		Gauss-Seidel		Jacobi	
N	Iter	Error	Iter	Error	Iter	Error
50	132	6.567×10^{-3}	2214	6.831×10^{-3}	4058	7.106×10^{-3}
75	193	2.898×10^{-3}	4467	3.471×10^{-3}	8931	3.470×10^{-3}
100	253	1.638×10^{-3}	7309	2.703×10^{-3}	13161	3.799×10^{-3}
300	732	2.001×10^{-4}	44205	1.046×10^{-2}	73738	2.304×10^{-2}

Tabla 4.2: Comparación del número de iteraciones Iter que requiere cada método iterativo, para alcanzar convergencia en la resolución del sistema tridiagonal (4.10) con una tolerancia de 10^{-5}.

	Thomas		SOR óptimo		Gauss-Seidel		Jacobi	
N	TCPU	Error	TCPU	Error	TCPU	Error	TCPU	Error
50	0.016	6.563×10^{-3}	0.016	6.567×10^{-3}	0.312	6.831×10^{-3}	0.172	7.106×10^{-3}
75	0.016	2.896×10^{-3}	0.031	2.898×10^{-3}	0.641	3.471×10^{-3}	0.375	3.470×10^{-3}
100	0.016	1.628×10^{-3}	0.047	1.638×10^{-3}	1.469	2.703×10^{-3}	1.188	3.799×10^{-3}
300	0.031	1.816×10^{-4}	0.328	2.001×10^{-4}	19.750	1.046×10^{-2}	10.562	2.304×10^{-2}

Tabla 4.3: Comparación del tiempo de CPU, en segundos, que requiere cada uno de los tres métodos iterativos y el método de Thomas, para resolver el sistema tridiagonal (4.10) con una tolerancia de 10^{-5}.

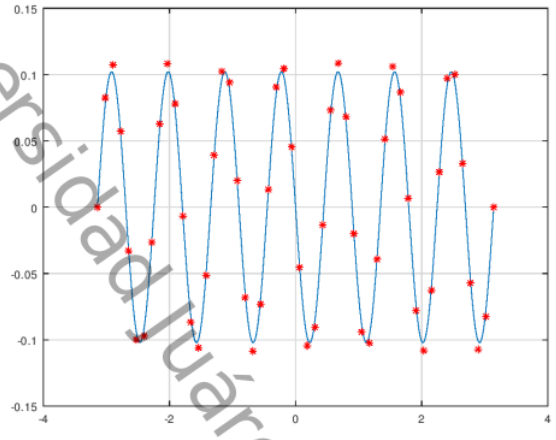


Figura 4.1: Aproximación numérica de la solución del problema de contorno (4.11) en 50 puntos uniformemente espaciados en el intervalo $(-\pi, \pi)$. La línea azul es la solución exacta dada en (4.12) y los punto rojos corresponden a la solución numérica.

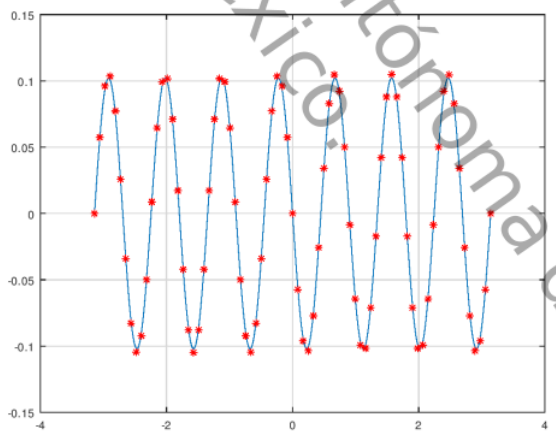


Figura 4.2: Aproximación numérica de la solución del problema de contorno (4.11) en 75 puntos uniformemente espaciados en el intervalo $(-\pi, \pi)$. La línea azul es la solución exacta dada en (4.12) y los punto rojos corresponden a la solución numérica.

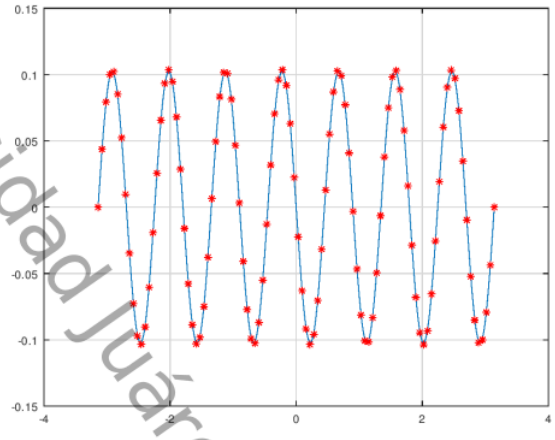


Figura 4.3: Aproximación numérica de la solución del problema de contorno (4.11) en 100 puntos uniformemente espaciados en el intervalo $(-\pi, \pi)$. La línea azul es la solución exacta dada en (4.12) y los punto rojos corresponden a la solución numérica.

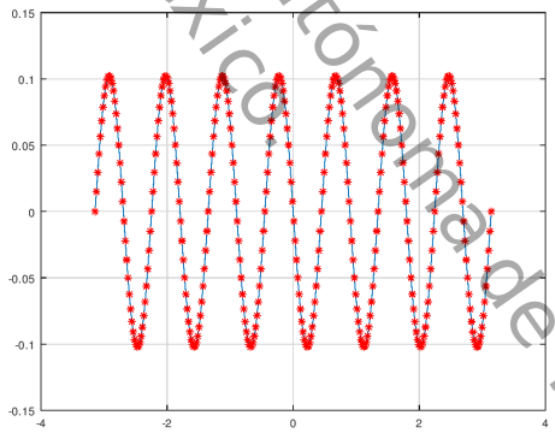


Figura 4.4: Aproximación numérica de la solución del problema de contorno (4.11) en 300 puntos uniformemente espaciados en el intervalo $(-\pi, \pi)$. La línea azul es la solución exacta dada en (4.12) y los punto rojos corresponden a la solución numérica.

4.3. Ecuación de Poisson en dos dimensiones

Abordaremos ahora el problema de contorno tipo Dirichlet de la ecuación de Poisson en dos dimensiones espaciales. Sea $\Omega = (0,1) \times (0,1)$. El problema lo formulamos como sigue. Dada $f : \overline{\Omega} \to \mathbb{R}$ continua, encontrar $u : \overline{\Omega} \to \mathbb{R}$ tal que

$$\left.\begin{array}{rcll} -u_{xx} - u_{yy} & = & f(x,y)\,, & (x,y) \in \Omega, \\ u(x,y) & = & 0, & (x,y) \in \partial\Omega. \end{array}\right\} \tag{4.13}$$

Puesto que varios métodos para la solución de problemas de valores de contorno, usualmente son comparados con este problema, el problema (4.13) también se le llama **problema modelo**. Para resolver (4.13), seguiremos la estrategia de la sección 4.2, aplicando el método de diferencias finitas para reemplazar el operador diferencial por un operador en diferencias. Para ello, se cubre $\Omega \cup \partial\Omega$ con una malla $\Omega_h \cup \partial\Omega_h$ como sigue:

$$\begin{aligned} \Omega_h &:= \{(x_i, y_j) \,|\, i,j = 1 : N\}\,, \\ \partial\Omega_h &:= \{(x_i, 0), (x_i, 1), (0, y_j), (1, y_j) \,|\, i,j = 0 : N+1\}\,, \end{aligned}$$

donde

$$x_i = ih\,, \quad y_j = jh, \quad i,j = 0 : N+1\,,$$
$$h := \frac{1}{N+1}\,, \quad N \geq 1 \text{ un número entero.}$$

En la figura 4.5(a) se ilustra geométricamente la construcción de esta malla, en particular, en la figura 4.5(b) se ilustra el caso cuando $N = 3$.

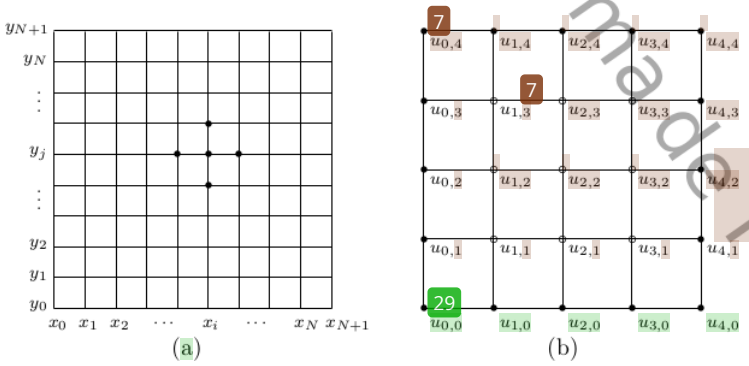


Figura 4.5: (a) malla cuadrangular $\Omega_h \cup \partial\Omega_h$ usada para la discretización de la ecuación de Poisson (4.13) en la región Ω. (b) valores de u conocidos en $\partial\Omega_h$ y valores de u desconocidos en Ω_h para $N = 3$.

Aplicando la fórmula de diferencia central (4.5) a la función u en el punto (x, y), resulta para el operador diferencial

$$\begin{aligned} -u_{xx} - u_{yy} = & -\frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} \\ & -\frac{u(x,y+h) - 2u(x,y) + u(x,y-h)}{h^2} + O(h^2) \\ = & \frac{4u(x,y) - u(x-h,y) - u(x+h,y) - u(x,y-h) - u(x,y+h)}{h^2} + O(h^2)\,. \end{aligned} \tag{4.14}$$

Si se abrevia

$$u_{i,j} := u(x_i, y_j)\,, \quad i,j = 0 : N+1,$$

resulta ahora que el operador diferencial $-u_{xx} - u_{yy}$ expresada en (4.14) puede ser reemplazado para todo $(x_i, y_j) \in \Omega_h$ por el operador en diferencias

$$\frac{4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} + \varepsilon_{ij}, \tag{4.15}$$

con un error de truncamiento $\varepsilon_{i,j}$ en cada punto $(x_i, y_j) \in \partial\Omega_h$.
Puesto que las condiciones de frontera en (4.13) imponen que $u_{i,j} = 0$ para $(x_i, y_j) \in \partial\Omega_h$, se sigue de (4.13) y (4.15) que los valores desconocidos $u_{i,j}$, $i,j = 1 : N$, deben satisfacer un sistema de ecuaciones lineales algebraicas de la forma

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f_{i,j} - h^2\varepsilon_{i,j}\,, \quad i,j = 1 : N, \tag{4.16}$$

donde $f_{i,j} = f(x_i, y_j)$, $(x_i, y_j) \in \Omega_h$.
Aquí los errores $\varepsilon_{i,j}$ dependen del tamaño h de la malla, y si la solución exacta u es suficientemente diferenciable, entonces $\varepsilon_{i,j} = O(h^2)$. Si se omite el error de truncamiento $\varepsilon_{i,j}$ en (4.16), resulta el sistema lineal de ecuaciones algebraicas:

$$\begin{aligned} & 4z_{i,j} - z_{i-1,j} - z_{i+1,j} - z_{i,j-1} - z_{i,j+1} = h^2 f_{i,j}\,, \quad i,j = 1 : N, \\ & z_{0,j} = z_{N+1,j} = z_{i,0} = z_{i,N+1} = 0 \quad \text{para} \quad i,j = 0 : N+1, \end{aligned} \tag{4.17}$$

que cuando h es suficientemente pequeño, se espera que cada componente z_{ij}, $i,j = 1 : N$, de la solución de (4.17) sea aproximadamente igual a $u_{i,j}$. En cada punto (x_i, y_j) de Ω_h le corresponde la componente $z_{i,j}$ de la solución de (4.17). Así que (4.17) es un sistema lineal de N^2 ecuaciones con N^2 incógnitas dadas por

$$z = (z_{1,1}, z_{2,1}, \ldots, z_{N,1}, z_{1,2}, z_{2,2}, \ldots, z_{N,2}, \ldots, z_{1,N}, z_{2,N}, \ldots, z_{N,N})^T.$$

Una forma conveniente de ordenar el sistema (4.17) es comenzar escribiendo la ecuación que le corresponde a la "primera fila interior" de la malla $\Omega_h \cup \partial\Omega_h$ (véase la figura 4.5(b) de abajo hacia arriba), es decir, la ecuación que le corresponde a (x_i, y_1), $i = 1 : N$; luego la ecuación que le corresponde a (x_i, y_2), $i = 1 : N$, y así sucesivamente hasta la última ecuación que

le corresponde a (x_i, y_N), $i = 1 : N$. Por ejemplo, para $N = 3$, y comenzando con $j = 1$, tenemos:

$$\begin{aligned} 4z_{1,1} - z_{0,1} - z_{2,1} - z_{1,0} - z_{1,2} &= h^2 f_{1,1} \\ 4z_{2,1} - z_{1,1} - z_{3,1} - z_{2,0} - z_{2,2} &= h^2 f_{2,1} \\ 4z_{3,1} - z_{2,1} - z_{4,1} - z_{3,0} - z_{3,2} &= h^2 f_{3,1} \end{aligned}$$

con $j = 2$:

$$\begin{aligned} 4z_{1,2} - z_{0,2} - z_{2,2} - z_{1,1} - z_{1,3} &= h^2 f_{1,2} \\ 4z_{2,2} - z_{1,2} - z_{3,2} - z_{2,1} - z_{2,3} &= h^2 f_{2,2} \\ 4z_{3,2} - z_{2,2} - z_{4,2} - z_{3,1} - z_{3,3} &= h^2 f_{3,2} \end{aligned}$$

y con $j = 3$:

$$\begin{aligned} 4z_{1,3} - z_{0,3} - z_{2,3} - z_{1,2} - z_{1,4} &= h^2 f_{1,3} \\ 4z_{2,3} - z_{1,3} - z_{3,3} - z_{2,2} - z_{2,4} &= h^2 f_{2,3} \\ 4z_{3,3} - z_{2,3} - z_{4,3} - z_{3,2} - z_{3,4} &= h^2 f_{3,3}\,. \end{aligned}$$

Recordando que $u_{i,j} = 0$ en $\partial\Omega_h$, el sistema lineal anterior se puede escribir en notación matricial como:

$$\begin{bmatrix} 4 & -1 & 0 & \vdots & -1 & 0 & 0 & \vdots & 0 & 0 & 0 \\ -1 & 4 & -1 & \vdots & 0 & -1 & 0 & \vdots & 0 & 0 & 0 \\ 0 & -1 & 4 & \vdots & 0 & 0 & -1 & \vdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & 0 & \vdots & 4 & -1 & 0 & \vdots & -1 & 0 & 0 \\ 0 & -1 & 0 & \vdots & -1 & 4 & -1 & \vdots & 0 & -1 & 0 \\ 0 & 0 & -1 & \vdots & 0 & -1 & 4 & \vdots & 0 & 0 & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & -1 & 0 & 0 & \vdots & 4 & -1 & 0 \\ 0 & 0 & 0 & \vdots & 0 & -1 & 0 & \vdots & -1 & 4 & -1 \\ 0 & 0 & 0 & \vdots & 0 & 0 & -1 & \vdots & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} z_{1,1} \\ z_{2,1} \\ z_{3,1} \\ z_{1,2} \\ z_{2,2} \\ z_{3,2} \\ z_{1,3} \\ z_{2,3} \\ z_{3,3} \end{bmatrix} = \begin{bmatrix} h^2 f_{1,1} \\ h^2 f_{2,1} \\ h^2 f_{3,1} \\ h^2 f_{1,2} \\ h^2 f_{2,2} \\ h^2 f_{3,2} \\ h^2 f_{1,3} \\ h^2 f_{2,3} \\ h^2 f_{3,3} \end{bmatrix}.$$

En general, tomando

$$b = h^2(f_{1,1}, f_{2,1}, \ldots, f_{N,1}, f_{1,2}, f_{2,2}, \ldots, f_{N,2}, \ldots, f_{1,N}, f_{2,N}, \ldots, f_{N,N})^T \in \mathbb{R}^{N^2},$$

la matriz de coeficientes del sistema lineal $Az = b$, toma la forma:

$$\begin{bmatrix}
4 & -1 & & & \vdots & -1 & & & \vdots & & & & \vdots & & & & & \\
-1 & \ddots & \ddots & & \vdots & & \ddots & & \vdots & & & & \vdots & & & & & \\
 & \ddots & \ddots & -1 & \vdots & & & \ddots & \vdots & & & & \vdots & & & & & \\
 & & -1 & 4 & \vdots & & & & -1 & \vdots & & & \vdots & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-1 & & & & \vdots & 4 & -1 & & \vdots & \ddots & & & \vdots & & & & & \\
 & \ddots & & & \vdots & -1 & \ddots & \ddots & \vdots & & \ddots & & \vdots & & & & & \\
 & & \ddots & & \vdots & & \ddots & \ddots & -1 & \vdots & & \ddots & \vdots & & & & & \\
 & & & -1 & \vdots & & -1 & 4 & \vdots & & & \ddots & \vdots & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & \vdots & \ddots & & & \vdots & \ddots & & & \vdots & -1 & & & & \\
 & & & & \vdots & & \ddots & & \vdots & & \ddots & & \vdots & & \ddots & & & \\
 & & & & \vdots & & & \ddots & \vdots & & & \ddots & \vdots & & & \ddots & & \\
 & & & & \vdots & & & & \ddots & \vdots & & & \ddots & \vdots & & & & -1 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & \vdots & & & & \ddots & -1 & & & \vdots & 4 & -1 & & & \\
 & & & & \vdots & & & & \vdots & \ddots & & & \vdots & -1 & \ddots & \ddots & & \\
 & & & & \vdots & & & & \vdots & & \ddots & & \vdots & & \ddots & \ddots & -1 & \\
 & & & & \vdots & & & & \vdots & & & -1 & \vdots & & & -1 & 4
\end{bmatrix}$$

Que es una matriz simétrica y de diagonal dominante, y se puede escribir como una **matriz tridiagonal por bloques**, como:

$$A = \begin{bmatrix} T_N & -I & & \\ -I & T_N & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & T_N \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2}, \tag{4.18}$$

donde

$$T_N = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix} \in \mathbb{R}^{N \times N}$$

es una matriz tridiagonal y estrictamente diagonal dominante de tamaño $N \times N$, e I es la matriz identidad también de tamaño $N \times N$. La matriz T_N aparece N veces en el bloque diagonal principal de A.

Es importante notar que la matriz A fue particionada de manera natural en bloques T_N de orden N, inducido por la forma de ordenar los puntos (x_i, y_j) de Ω_h en filas horizontales $(x_1, y_j), (x_2, y_j), \ldots, (x_N, y_j)$.
La matriz A es *sparse*, en cada fila hay a los más cinco elementos diferentes de cero. Por esta razón, en la ejecución de un paso de iteración $z^{(k)} \to z^{(k+1)}$, el método de Jacobi o el método de Gauss-Seidel, requieren únicamente alrededor de $5N^2$ operaciones o flops (1 operación = 1 multiplicación o división +1 adición). Si la matriz A fuera densa, estos métodos necesitarían N^4 operaciones en cada paso [9]. Comparemos este coste con el de un método directo para resolver $Az = b$. Como A es simétrica y se puede probar que también es definida positiva, entonces el sistema se podría resolver con el método de Cholesky. Para ello, se necesita calcular la descomposición triangular $A = LL^T$, donde L sería una matriz triangular inferior de $N^2 \times N^2$ de la forma:

$$L = \underbrace{\begin{bmatrix} x & & & & \\ \vdots & \ddots & & & \\ x & \cdots & \ddots & & \\ & \ddots & \cdots & \ddots & \\ & & x & \cdots & x \end{bmatrix}}_{N+1}$$

El cálculo de L solo requiere de aproximadamente $\frac{1}{2}N^4$ operaciones. Puesto que el método de Jacobi, por ejemplo, requiere aproximadamente de $(N+1)^2$ iteraciones (el método de SOR requiere de $N+1$ iteraciones), véase (4.23), para obtener un resultado exacto de 2 decimales, el método de Cholesky sería menos costoso que el método de Jacobi. Sin embargo, la principal dificultad con el método de Cholesky, radica en el hecho de que necesita almacenar aproximadamente N^3 elementos no nulos de L que ocuparía mucho espacio de memoria (el orden de magnitud típico de N es entre 50 y 100). Es aquí donde se manifiestan las ventajas de los métodos iterativos: su requisito de almacenamiento es solo del orden de magnitud N^2 [9].
Como corolario de los resultados revisados en la sección 2.5 del Capítulo 2, tenemos que la matriz A tridiagonal por bloques dada en (4.18), tiene excelentes propiedades.

Lema 4.3.1 (Stoer and Bulirsch (1993) [9]). *Sea A la matriz tridiagonal por bloques dada en* (4.18)*, $A = D - L - U$ como en* (2.1) *y $E_J = D^{-1}(L+U)$ como en* (2.3)*. Entonces*

a) $E_J = \frac{1}{4}(4I - A)$.

b) A tiene la propiedad A.

c) A es coherentemente ordenada.

Demostración. Se puede consultar en el libro de Stoer and Bulirsch (1993) [9]. □

Los valores propios y vectores propios de la matriz de iteración E_J, del método de Jacobi correspondiente, se muestran en el lema 4.3.2.

Lema 4.3.2 (Stoer and Bulirsch (1993) [9]). *Sea A la matriz tridiagonal por bloques dada en (4.18), $A = D - L - U$ como en (2.1) y $E_J = D^{-1}(L+U)$ como en (2.3). Entonces*

a) Los vectores propios $z^{(k,l)}$, $k = 1:N$, $l = 1:N$, de E_J, son

$$z_{i,j}^{(k,l)} = \operatorname{sen}\frac{k\pi i}{N+1}\operatorname{sen}\frac{l\pi j}{N+1}, \quad i,j = 1:N.$$

b) Los valores propios $\mu^{(k,l)}$, $k = 1:N$, $l = 1:N$, de E_J son reales y están dados por

$$\mu^{(k,l)} = \frac{1}{2}\left(\cos\frac{k\pi}{N+1} + \cos\frac{l\pi}{N+1}\right), \quad k,l = 1:N.$$

c) El radio espectral de E_J es

$$\rho(E_J) = \max_{k,l=1:N}\left|\mu^{(k,l)}\right| = \cos\frac{\pi}{N+1} < 1, \quad N \geq 2. \tag{4.19}$$

Demostración. Cada inciso se verifica por sustitución directa. Más detalles en Stoer and Bulirsch (1993) [9]. □

Como consecuencias del corolario 2.6.1 y del teorema 2.6.5, tenemos el siguiente corolario sobre los métodos de Gauss-Seidel y SOR.

Corolario 4.3.1 (Stoer and Bulirsch (1993) [9]). *Sea A la matriz tridiagonal por bloques dada en (4.18), $A = D - L - U$ como en (2.1), $E_{GS} = (D-L)^{-1}U$ como en (2.6) y E_ω como en (2.20). Entonces*

a) El radio espectral de la matriz de iteración del método de Gauss-Seidel E_{GS} es

$$\rho(E_{GS}) = \cos^2\frac{\pi}{N+1} < 1, \quad N \geq 2.$$

b) El parámetro de relajación óptimo para la convergencia del método de SOR es

$$\omega_b = \frac{2}{1 + \operatorname{sen}\frac{\pi}{N+1}}.$$

c) El radio espectral de la matriz de iteración de E_{ω_b} es

$$\rho(E_{\omega_b}) = \frac{\cos^2\frac{\pi}{N+1}}{\left(1 + \operatorname{sen}\frac{\pi}{N+1}\right)^2}.$$

El número $\kappa = \kappa(N)$ con $[\rho(E_J)]^\kappa = \rho(E_{\omega_b})$ indica cuántos pasos del método de Jacobi producen la misma reducción de error que un paso del método de relajación óptimo. Es claro que

$$\kappa = \frac{\ln \rho(E_{\omega_b})}{\ln \rho(E_J)}. \tag{4.20}$$

Puesto que para z pequeño,

$$\ln(1+z) = z - \frac{z^2}{2} + O(z^3),$$

y para N grande:

$$\rho(E_J) = \cos\frac{\pi}{N+1} = 1 - \frac{\pi^2}{2(N+1)^2} + O\left(\frac{1}{N^4}\right),$$

y

$$\operatorname{sen}\frac{\pi}{N+1} = \frac{\pi}{N+1} + O\left(\frac{1}{N^3}\right).$$

Entonces

$$\begin{aligned}
\ln \rho(E_J) &= \ln\left(1 - \frac{\pi^2}{2(N+1)^2} + O\left(\frac{1}{N^4}\right)\right) \\
&= -\frac{\pi^2}{2(N+1)^2} + O\left(\frac{1}{N^4}\right) - \frac{1}{2}\left(-\frac{\pi^2}{2(N+1)^2} + O\left(\frac{1}{N^4}\right)\right)^2 + O\left(\frac{1}{N^{12}}\right) \\
&= -\frac{\pi^2}{2(N+1)^2} + O\left(\frac{1}{N^4}\right).
\end{aligned} \tag{4.21}$$

Por otro lado, se sigue del corolario 4.3.1(c), que

$$\begin{aligned}
\ln \rho(E_{\omega_b}) &= 2\left[\ln \rho(E_J) - \ln\left(1 + \operatorname{sen}\frac{\pi}{N+1}\right)\right] \\
&= 2\left[-\frac{\pi^2}{2(N+1)^2} - \operatorname{sen}\frac{\pi}{N+1} + \frac{1}{2}\operatorname{sen}^2\frac{\pi}{N+1} + O\left(\frac{1}{N^4}\right)\right] \\
&= 2\left[-\frac{\pi^2}{2(N+1)^2} - \frac{\pi}{N+1} + \frac{1}{2}\frac{\pi^2}{(N+1)^2} + O\left(\frac{1}{N^3}\right)\right] \\
&= -\frac{2\pi}{N+1} + O\left(\frac{1}{N^3}\right).
\end{aligned} \tag{4.22}$$

Finalmente, asintóticamente para N grande, resulta de (4.20), (4.21) y (4.22), que

$$\kappa = \kappa(N) \cong \frac{4(N+1)}{\pi},$$

que nos dice que el método de relajación óptimo es más de N veces más rápido que el método de Jacobi. Las cantidades

$$\left.\begin{aligned} R_J &:= -\frac{\ln 10}{\ln \rho(E_J)} \cong 0.467(N+1)^2, \\ R_{GS} &:= -\frac{1}{2}R_J \cong 0.234(N+1)^2, \\ R_{w_b} &:= -\frac{\ln 10}{\ln \rho(E_{w_b})} \cong 0.367(N+1), \end{aligned}\right\} \tag{4.23}$$

indican el número de iteraciones que requieren el método de Jacobi[5], el método de Gauss-Seidel y el método de relajación óptimo, respectivamente, para reducir el error por un factor de $\frac{1}{10}$, [9].

4.4. Ejemplo numérico

En esta última sección, presentamos la solución numérica del problema de contorno tipo Dirichlet de la ecuación de Poisson en dos dimensiones espaciales, formulada como sigue [9]: encontrar $u : \overline{\Omega} \to \mathbb{R}$ tal que

$$\left.\begin{aligned} -u_{xx} - u_{yy} &= 2\pi^2 \operatorname{sen}(\pi x)\operatorname{sen}(\pi y), \quad (x,y) \in \Omega \subset \mathbb{R}^2, \\ u(x,y) &= 0, \quad (x,y) \in \partial\Omega, \end{aligned}\right\} \tag{4.24}$$

donde $\Omega = (0,1) \times (0,1)$.
La solución exacta está dada por

$$u(x,y) = \operatorname{sen}(\pi x)\operatorname{sen}(\pi y), \quad (x,y) \in \overline{\Omega}.$$

Usando la discretización descrita en la sección 4.3, resulta el sistema tridiagonal por bloques de ecuaciones lineales

$$\begin{aligned} Az &= b, \quad A \text{ como en (4.18)}, \\ b &= 2h^2\pi^2\overline{u}, \end{aligned} \tag{4.25}$$

donde,

$$\begin{aligned} \overline{u} &= (\overline{u}_{1,1}, \overline{u}_{2,1}, \ldots, \overline{u}_{N,1}, \overline{u}_{1,2}, \overline{u}_{2,2}, \ldots, \overline{u}_{N,2}, \ldots, \overline{u}_{1,N}, \overline{u}_{2,N}, \ldots, \overline{u}_{N,N})^T, \\ \overline{u}_{i,j} &= \operatorname{sen}(i\pi h)\operatorname{sen}(j\pi h), \quad h = \frac{1}{N+1}, \quad N \geq 1. \end{aligned}$$

[5]Notemos que si k es el número de iteraciones que requiere el método de Jacobi para reducir el error en un factor de $\frac{1}{10}$, entonces $[\rho(E_J)]^k = \frac{1}{10}$.

Ahora aproximaremos valores de u en los N^2 puntos (x_i, y_j) de Ω_h, resolviendo el sistema tridiagonal por bloques (4.25). Como aplicaremos los métodos iterativos de Jacobi, Gauss-Seidel y relajación, para resolver dicho sistema; calcularemos primero el ω óptimo que garantice una mayor velocidad de convergencia para el método de relajación, para cada N. Para ello, aplicaremos el lema 4.3.2 y el corolario 4.3.1. Los valores óptimos obtenidos de ω, así como los radios espectrales de las matrices de iteración de los métodos iterativos, se presentan en la tabla 4.4. Es importante notar que los valores de N que se obtuvieron para 50, 75 y 100 puntos, son exactamente los mismos, que se obtuvieron en la tabla 4.1.

N	$\rho(E_J)$	$\rho(E_{GS})$	ω_b	$\rho(E_{\omega_b})$
50	0.99810	0.99621	1.8840	0.88402
75	0.99915	0.99829	1.9206	0.92063
100	0.99952	0.99903	1.9397	0.93968
125	0.99969	0.99938	1.9514	0.95135

Tabla 4.4: Valores óptimos ω_b para la convergencia del método de SOR y radios espectrales.

En la tabla 4.5 se presenta una comparación entre los métodos iterativos de SOR, Gauss-Seidel y Jacobi, en la solución numérica del sistema tridiagonal por bloques (4.25). Se tomó como punto de inicio a $x^{(0)} = (0, 0, \ldots, 0)^T \in \mathbb{R}^{N^2}$ en cada uno de ellos. Notamos que el método de SOR resultó más eficiente como era de esperarse, al requerir menor cantidad de iteraciones (Iter) y tiempo de CPU (en segundos) para alcanzar convergencia a la tolerancia 10^{-5}. El error que se reporta es el error absoluto máximo, es decir,

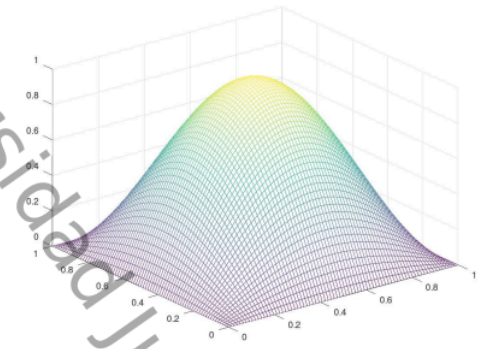
$$\text{Error} = \max_{1 \le i,j \le N} |u(x_i, y_j) - z_{i,j}|.$$

En la figura 4.6(a), se reporta la gráfica de la solución exacta $u(x, y) = \text{sen}(\pi x)\,\text{sen}(\pi y)$ del problema de contorno (4.24); y en la figura 4.6(b), se reporta la gráfica de la solución numérica $z = (z_{1,1}, z_{2,1}, \ldots, z_{N,N})^T$ con $N = 75$, obtenida al resolver el sistema (4.25) con una tolerancia de 10^{-5}. Finalmente, en la figura 4.6(c), se reporta la gráfica del error absoluto:

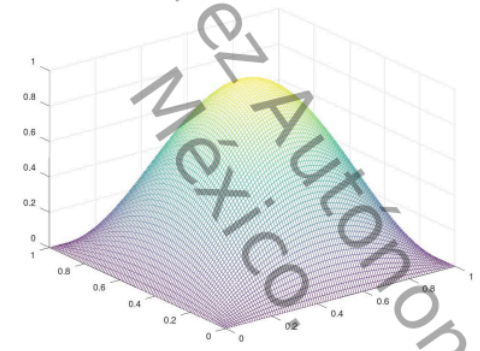
$$\text{Error absoluto} = |u(x_i, y_j) - z_{i,j}|, \quad i, j = 1 : N. \tag{4.26}$$

	SOR			Gauss-Seidel			Jacobi		
N	Iter	TCPU	Error	Iter	TCPU	Error	Iter	TCPU	Error
50	107	5.23	2.084×10^{-4}	1567	70.39	2.307×10^{-3}	2767	76.18	4.922×10^{-3}
75	154	35.11	9.893×10^{-5}	3013	671.22	5.672×10^{-3}	5219	666.44	1.143×10^{-2}
100	203	138.91	9.435×10^{-5}	4738	3203.31	1.014×10^{-2}	8061	3185.73	2.016×10^{-2}
125	253	813.63	1.029×10^{-4}	6670	19400.14	1.577×10^{-2}	11158	10243.88	3.112×10^{-2}

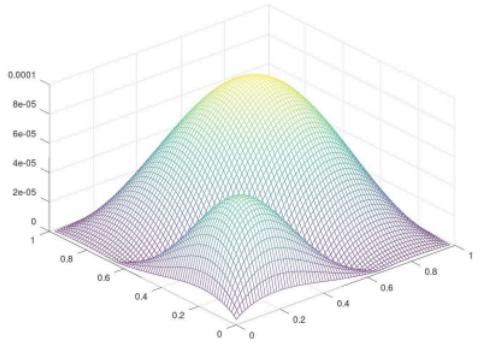
Tabla 4.5: Comparación del número de iteraciones Iter y del tiempo de CPU (en segundos), que requiere cada uno de los tres métodos iterativos para resolver el sistema tridiagonal por bloques (4.25) con una tolerancia de 10^{-5}.



(a) Solución exacta.



(b) Solución numérica.



(c) Error absoluto.

Figura 4.6: La solución numérica se obtuvo con el método de relajación con $\omega_b = 1.9206$, $Tol = 10^{-5}$ y $N = 75$. El error absoluto máximo es 9.893×10^{-5}.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Existen diferentes métodos analíticos para resolver la ecuación de Poisson, sin embargo se volvería una tarea difícil depender sólo de ellos para hallar la solución. Afortunadamente se cuenta con métodos numéricos que ayudan a encontrar una solución aproximada.

El método de Diferencias Finitas es uno de los clásicos: se discretiza el problema para calcular una solución aproximada en un número finito de puntos uniformemente espaciados y esto nos lleva a sustituir la ecuación diferencial por un sistema de ecuaciones lineales algebraicas y no importa si la ecuación de Poisson tiene una o más dimensiones.

Con el método de Diferencias Finitas obtenemos un sistema de ecuaciones lineales algebraicas, por lo que ahora nuestro problema será hallar la solución a dicho sistema. Tal tarea se lleva a cabo por medio de los métodos iterativos de Jacobi, Gauss-Seidel y SOR junto con el método de Thomas (un método directo).

Los experimentos numéricos fueron realizados en un computador portatil intel(R) Core(TM) i5-9300H CPU 2.40GHz, memoria RAM de 8 Gb y sistema operativo Windows 10 de 64 bits. Al comparar los cuatro métodos usados para hallar la solución del sistema de ecuaciones lineales algebraicas obtenidas de la ecuación de Poisson en 1D, podemos notar que el método de Thomas converge más rápido a la solución que los métodos iterativos usados en 50, 75, 100 y 300 puntos. En el caso de los métodos iterativos, el método de SOR, con un ω óptimo, converge más rápido a la solución que los métodos de Gauss-Seidel y Jacobi. El error absoluto del método de Thomas y el de SOR son muy cercanos.

Para el caso de la ecuación de Poisson en 2D, sólo se comparan los métodos iterativos, ya que la matriz obtenida con el método de Diferencias Finitas es una matriz tridiagonal por

bloques y el algoritmo de Thomas que hemos usado sólo funciona para matrices tridiagonales. Podemos observar que el método de SOR converge más rápido a la solución, en iteraciones y tiempo, que los métodos de Gauss-Seidel y Jacobi en 50, 75, 100 y 125 puntos. Ya no se trabaja con 300 puntos porque supera la capacidad de memoria de la computadora portatil utilizada. Notemos que con el método de SOR, en 125 puntos, incrementa el error absoluto máximo a comparación con 100 puntos. Aunque el método de SOR converge más rápido con 75 puntos que con 100 puntos, es con éste último donde se consigue el menor error absoluto posible.

5.2. Trabajos futuros

Estudiar métodos fuertes, como:

1. Los métodos iterativos por bloques.
2. El método ADI de Peaceman y Rachford, para abordar el problema de contorno:

$$\begin{aligned} -u_{xx} - u_{yy} + \sigma u &= f(x,y), \quad (x,y) \in \Omega \\ u(x,y) &= 0, \quad (x,y) \in \partial\Omega, \end{aligned} \tag{5.1}$$

   donde $\Omega = \{(x,y) | 0 < x < 1,\ 0 < y < 1\} \subset \mathbb{R}^2$ con frontera $\partial\Omega$.
3. El método del gradiente conjugado de Hestenes y Stiefel.
4. El algoritmo de Buneman para la solución de la ecuación de Poisson (5.1), pero ahora sobre $\Omega = \{(x,y) | 0 < x < a,\ 0 < y < b\} \subset \mathbb{R}^2$.
5. Métodos multigrid.
6. Comparación de los métodos iterativos.

Bibliografía

[1] Burden, R.L. y Faires, J.D. (1998). *Análisis Numérico*. Sexta Edición. México: International Thomson Editores.

[2] Demmel, J.W. (1997). *Applied Numerical Lineal Algebra*. Philadelphia: SIAM.

[3] Gover, M.J.C. (1994). *The eigenproblem of a tridiagonal 2-Toeplitz matrix*. Linear Algebra and its Applications 197, 198: 63–78. *https://doi.org/10.1016/0024-3795(94)90481-2*

[4] Isaacson, E. and Keller, B.H. (1966). *Analysis of Numerical Methods*. New York: Wiley.

[5] Ortega, J.M. (1972). *Numerical Analysis: A Second Course*. New York: Academic Press.

[6] Quarteroni, A., Sacco, R. and Saleri, F. (2007). *Numerical Mathematics*. New York: Springer-Verlag.

[7] Tijonov, A.N. y Samarsky, A.A. (1980). *Ecuaciones de la Física Matemática*. Moscú: MIR.

[8] Skiva, Y.N. (2005). *Métodos y Esquemas Numéricos: Un análisis computacional*. México: UNAM. *https://www.researchgate.net/publication/236170*

[9] Stoer, J. and Bulirsch, R. (1993). *Introduction to Numerical Analysis*. New York: Springer-verlag.

[10] Süli, E. and Mayers, D.F. (2003). *An Introduction to Numerical Analysis*. Cambridge University Press.

[11] Varga, R.S. (1962). *Matrix Iterative Analysis*. New Jersey: Prentice-Hall.

[12] Young, D.M. (1971). *Iterative Solution of Large Linear Systems*. Nueva York: Academic Press.

RESOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE POISSON EN 1D Y 2D POR EL MÉTODO DE DIFERENCIAS FINITAS

INFORME DE ORIGINALIDAD

6%
ÍNDICE DE SIMILITUD

FUENTES PRIMARIAS

1	repobib.ubiobio.cl Internet	213 palabras — 1%
2	hdl.handle.net Internet	110 palabras — 1%
3	dspace.ups.edu.ec Internet	63 palabras — < 1%
4	www.coursehero.com Internet	43 palabras — < 1%
5	www.scribd.com Internet	39 palabras — < 1%
6	idoc.pub Internet	36 palabras — < 1%
7	marian.fsik.cvut.cz Internet	33 palabras — < 1%
8	www.unalmed.edu.co Internet	33 palabras — < 1%
9	(5-6-03) http://193.144.183.17/~cordon/numeri98/sistline/html/lista09-0.html Internet	32 palabras — < 1%
10	id.scribd.com	

	Internet	31 palabras — < 1%
11	noexperiencenecessarybook.com Internet	31 palabras — < 1%
12	archive.org Internet	30 palabras — < 1%
13	www.ci2ma.udec.cl Internet	30 palabras — < 1%
14	studfile.net Internet	27 palabras — < 1%
15	mafiadoc.com Internet	26 palabras — < 1%
16	ri.ujat.mx Internet	23 palabras — < 1%
17	vdocumento.com Internet	23 palabras — < 1%
18	Anderlini, L.. "Social memory, evidence, and conflict", Review of Economic Dynamics, 201007 Crossref	22 palabras — < 1%
19	pubs.usgs.gov Internet	22 palabras — < 1%
20	"Cálculo Cientifico con MATLAB y Octave", Springer Science and Business Media LLC, 2006 Crossref	20 palabras — < 1%
21	doku.pub Internet	20 palabras — < 1%
22	www.passeidireto.com Internet	

		20 palabras — < 1%
23	www.mat.ufpr.br Internet	18 palabras — < 1%
24	Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, Paola Gervasio. "Matematica Numerica", Springer Nature, 2014 Crossref	15 palabras — < 1%
25	cimec.org.ar Internet	14 palabras — < 1%
26	Chia-Shin Chung. "A single-period inventory placement problem for a serial supply chain", Naval Research Logistics, 09/2001 Crossref	13 palabras — < 1%
27	ora.ox.ac.uk Internet	13 palabras — < 1%
28	David H. Eberly. "Game Physics", CRC Press, 2019 Publicaciones	12 palabras — < 1%
29	arxiv.org Internet	12 palabras — < 1%
30	docplayer.es Internet	12 palabras — < 1%
31	dspace.umh.es Internet	12 palabras — < 1%
32	export.arxiv.org Internet	12 palabras — < 1%
33	www.de.unifi.it Internet	12 palabras — < 1%

34	www.researchgate.net Internet	12 palabras — < 1%
35	patents.google.com Internet	11 palabras — < 1%
36	smb.org.mx Internet	11 palabras — < 1%
37	www.aviladigital.com Internet	11 palabras — < 1%
38	www.ugr.es Internet	11 palabras — < 1%
39	M.J.C. Gover. "The eigenproblem of a tridiagonal 2-Toeplitz matrix", Linear Algebra and its Applications, 1994 Crossref	10 palabras — < 1%
40	Molina Castro, Juan David. "Mecanismos para la inversion y remuneracion de la transmision de energia electrica", Pontificia Universidad Catolica de Chile (Chile), 2021 ProQuest	10 palabras — < 1%
41	docslib.org Internet	10 palabras — < 1%
42	edoc.pub Internet	10 palabras — < 1%
43	pastebin.com Internet	10 palabras — < 1%
44	sites.google.com Internet	10 palabras — < 1%
45	vdocuments.site Internet	10 palabras — < 1%

46	www.degruyter.com Internet	10 palabras — < 1%
47	www.regnumchristi.org Internet	10 palabras — < 1%
48	zaguan.unizar.es Internet	10 palabras — < 1%

EXCLUIR CITAS	ACTIVADO	EXCLUIR FUENTES	DESACTIVADO
EXCLUIR BIBLIOGRAFÍA	ACTIVADO	EXCLUIR COINCIDENCIAS	< 10 PALABRAS