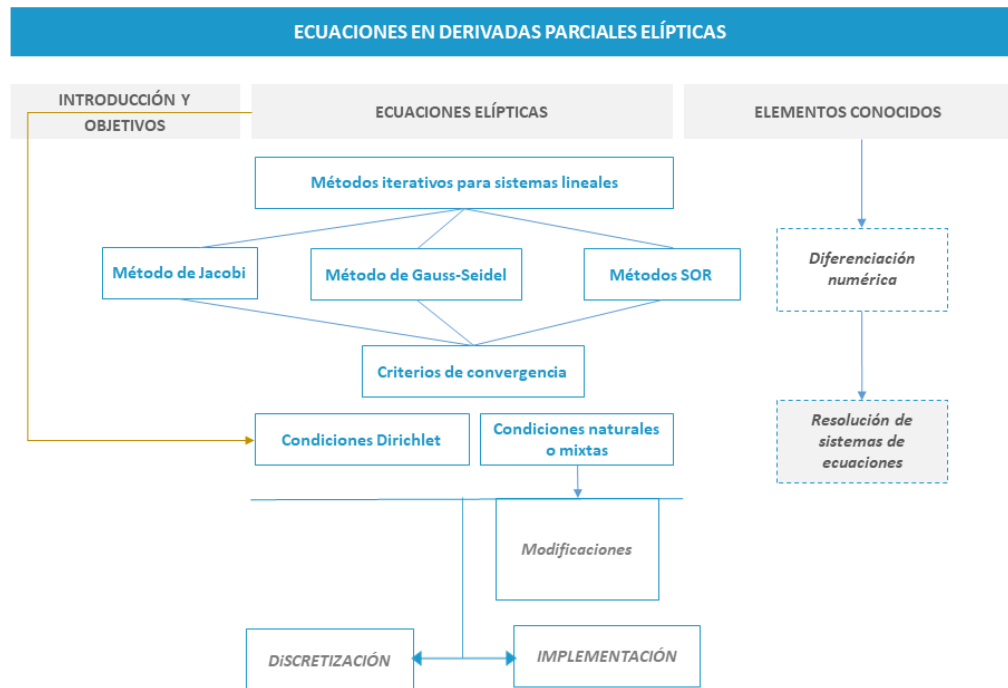


Métodos Numéricos II

Ecuaciones en derivadas parciales elípticas

Índice

Esquema.	2
Ideas clave	3
8.1 Introducción y objetivos	3
8.2 Sistemas de ecuaciones lineales	5
8.3 Problemas de contorno elípticos	28
8.4 Ejercicios propuestos	42



8.1 Introducción y objetivos

En los temas anteriores hemos aproximado la solución de problemas modelizados mediante ecuaciones en derivadas parciales parabólicas e hiperbólicas, utilizando la técnica de diferencias finitas, con diferentes tipos de condiciones de contorno: de tipo Dirichlet, naturales o mixtas.

En este tema, vamos a aplicar de nuevo la técnica de diferencias finitas para aproximar la solución de ecuaciones elípticas. Las ecuaciones de este tipo surgen de manera natural en el estudio de problemas físicos independientes del tiempo (por lo que no aparecerán condiciones iniciales), como por ejemplo, el estado estacionario de la distribución de calor en una región plana, la energía potencial de un punto en el plano bajo la acción de fuerzas gravitacionales y problemas estacionarios acerca de fluidos incompresibles.

Como vimos en el Tema 4, dentro de las ecuaciones en derivadas parciales lineales de segundo orden,

$$A(x, y)u_{xx} + B(x, y)u_{xy} + C(x, y)u_{yy} + D(x, y)u_x + E(x, y)u_y + F(x, y)u + G(x, y) = 0,$$

donde A, B, C, D, E, F y G son, en principio, funciones cualesquiera de dos variables; las ecuaciones elípticas deben cumplir

$$\Delta = B^2(x, y) - A(x, y)C(x, y) < 0, \quad \forall x, y.$$

La descripción más sencilla de un problema de este tipo es

$$\nabla^2 u(x, y) \equiv u_{xx}(x, y) + u_{yy}(x, y) = f(x, y), \quad (x, y) \in R,$$

$$u(x, y) = g(x, y) \text{ para } (x, y) \in S,$$

donde $R = \{(x, y) : a < x < b, c < y < d\}$ y S es la frontera de R .

Mediante diferencias finitas transformamos el problema en un sistema de ecuaciones lineales (ya que la EDP es lineal) cuya solución nos dará valores aproximados del problema de contorno en los puntos elegidos. Al igual que en los otros tipos de ecuaciones en derivadas parciales, las condiciones de frontera pueden ser de tipo Dirichlet, naturales o mixtas. Las diferencias finitas aplicadas a problemas elípticos van a dar siempre métodos implícitos.

En los métodos implícitos para problemas parabólicos o hiperbólicos, encontrábamos la solución en cada instante temporal resolviendo un sistema cuyo tamaño coincidía con el número de nodos espaciales en los que no se conocía la solución, lo que daba lugar a sistemas de tamaño razonable. En el caso de las ecuaciones elípticas, los sistemas que van a aparecer serán de gran tamaño, ya que el número de ecuaciones e incógnitas coincidirá con el producto del número de nodos del intervalo $[a, b]$ con el número de nodos del intervalo $[c, d]$ en los que no se conozca la solución.

Frente al método de eliminación de Gauss para resolver un sistema lineal, es más útil utilizar métodos iterativos para resolver estos sistemas de gran tamaño, ya que se disminuye el coste computacional y en consecuencia, el error de redondeo.

Los objetivos que perseguimos en el presente tema son los siguientes:

► Descripción de los métodos iterativos para sistemas lineales.

- Método de Jacobi
- Método de Gauss-Seidel

- Métodos SOR
- ▶ Criterios de convergencia para los métodos iterativos
- ▶ Implementación y aplicación de estos métodos
- ▶ Introducción a las EDPs elípticas
 - Transformación de la EDP en una ecuación en diferencias
 - Aplicación de los métodos iterativos a la resolución de estos problemas
 - Implementación en Matlab
- ▶ Problemas elípticos con condiciones no Dirichlet. Modificación de los programas

8.2 Sistemas de ecuaciones lineales

En numerosos problemas de ciencia e ingeniería aparece la necesidad de resolver sistemas lineales $Ax = b$, donde A es una matriz cuadrada de tamaño $n \times n$ y b un vector real o complejo de n componentes. Los métodos de resolución de este tipo de sistemas se pueden clasificar de la siguiente forma:

- ▶ Métodos directos
 - Si A es invertible, $x = A^{-1}b$
 - Método de Cramer
 - Método de eliminación de Gauss con sus múltiples variantes: pivotación parcial, factorización LU , ...
- ▶ Métodos iterativos, $x = Hx + d$, H matriz $n \times n$, $d \in \mathbb{R}^n$
 - Métodos iterativos estacionarios: Jacobi, Gauss-Seidel, ...
 - Métodos de direcciones alternadas
 - Métodos de gradiente conjugado

► Métodos que utilizan preconditionadores

Los sistemas a los que queremos dedicar este tema y que aparecen al discretizar una ecuación en derivadas parciales elíptica, presentan dos características esenciales:

- El sistema es de gran tamaño, n es muy grande
- La matriz de coeficientes es dispersa, es decir, un número de elementos no nulos, $nnz(A)$, del orden $nnz(A) = cn$, con c independiente de n . Además, no presenta una estructura clara

Estas características desaconsejan el uso de métodos directos, ya que

- El orden de magnitud del número de operaciones para calcular A^{-1} es $O(n^3)$, lo que en tiempo de ejecución lo haría inviable
- Tanto el cálculo de A^{-1} como el método de eliminación de Gauss hace perder el carácter disperso de la matriz A , lo que se traduce en un mayor número de operaciones y en el incremento del error de redondeo

Debemos, por tanto, recurrir a los METODOS ITERATIVOS, a los que vamos a prestar atención en la primera parte de este tema.

Conceptos básicos

Un método iterativo obtiene una solución aproximada del sistema $Ax = b$ construyendo una sucesión de vectores

$$x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots \text{ en } \mathbb{R}^n$$

a partir de un vector arbitrario $x^{(0)}$ que denominamos aproximación inicial.

Un método iterativo se dice que es convergente si

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

donde \bar{x} denota la solución del sistema $Ax = b$. Dos vectores van a jugar un papel importante en estos procesos:

- El vector error, en cada iteración, que se define como

$$e_k = \bar{x} - x^{(k)},$$

- El vector residuo, en cada iteración, que se define como

$$r_k = b - Ax^{(k)}.$$

Es sencillo demostrar el siguiente resultado:

$$\lim_{k \rightarrow \infty} x^{(k)} = \bar{x} \Leftrightarrow \lim_{k \rightarrow \infty} \|e_k\| = 0 \Leftrightarrow \lim_{k \rightarrow \infty} \|r_k\| = 0,$$

donde $\|\cdot\|$ representa cualquier norma vectorial.

Los métodos directos, teóricamente, dan la solución exacta; pero en un ordenador pueden dar lugar a numerosos errores de redondeo que hagan inservible el resultado obtenido. Por su parte, un método iterativo nunca da la solución exacta incluso trabajando en precisión infinita. Debemos establecer un criterio de parada para que el ordenador deje de calcular iterados. En general se da a priori una precisión Tol del error máximo que estamos dispuestos a asumir. El criterio de parada perfecto sería aquel que hiciese detenerse al ordenador cuando

$$\|e_k\| < Tol \text{ (error absoluto)} \quad \text{ó} \quad \frac{\|e_k\|}{\|\bar{x}\|} < Tol \text{ (error relativo)}.$$

Pero, \bar{x} y e_k no son conocidos, luego este criterio no es útil.

Debemos recurrir al criterio del residuo ó al criterio de la diferencia de los dos últimos iterados

$$\|r_k\| < Tol \text{ (error absoluto)} \quad \text{ó} \quad \frac{\|r_k\|}{\|b\|} < Tol \text{ (error relativo),}$$

es decir,

$$\|b - Ax^{(k)}\| < Tol \text{ (error absoluto)} \quad \text{ó} \quad \frac{\|b - Ax^{(k)}\|}{\|b\|} < Tol \text{ (error relativo),}$$

o bien

$$\|x^{(k+1)} - x^{(k)}\| < Tol.$$

Podemos establecer la siguiente relación entre el error y el residuo

$$r_k = b - Ax^{(k)} = A\bar{x} - Ax^{(k)} = Ae_k.$$

Así, utilizando cualquier norma matricial

$$\|r_k\| \leq \|A\|\|e_k\|; \quad \|e_k\| \leq \|A^{-1}\|\|r_k\|.$$

Además,

$$\|\bar{x}\| \leq \|A^{-1}\|\|b\|; \quad \|b\| \leq \|A\|\|A^{-1}b\| = \|A\|\|\bar{x}\|,$$

y combinando ambas desigualdades, obtenemos

$$\frac{1}{\|A\|\|A^{-1}\|} \frac{\|r_k\|}{\|b\|} \leq \frac{\|e_k\|}{\|\bar{x}\|} \leq \|A\|\|A^{-1}\| \frac{\|r_k\|}{\|b\|}$$

Teniendo en cuenta que $\mathcal{K}(A) = \|A\|\|A^{-1}\|$ es el número de condición de la matriz A , podemos afirmar:

El test del residuo es fiable si $\mathcal{K}(A)$ no es muy grande, es decir, A es una **matriz estable o bien condicionada**.

Nos vamos a centrar a continuación en los denominados **métodos iterativos estacio-**

narios.

Sea A la matriz del sistema $Ax = b$. Podemos considerar la partición (también llamada splitting)

$$A = M - N,$$

donde $M \neq A$ es una matriz invertible. Construimos el proceso iterativo

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b = Hx^{(k)} + q, \quad k = 0, 1, \dots,$$

donde H es la matriz de iteración y $x^{(0)}$ la aproximación inicial. Esto es equivalente a

$$x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}) = x^{(k)} + M^{-1}r_k.$$

Diremos que un método iterativo es **estacionario** si la matriz de iteración H es constante en todo el proceso.

Las condiciones para que un método iterativo estacionario sea convergente van a estar asociadas con condiciones sobre la matriz del sistema A o sobre la matriz de iteración H , que suele ser distinta para cada método iterativo. Veamos, a modo de ejemplo, algunos resultados.

Teorema 1

Consideremos un método estacionario con matriz H . Si $\|H\| < 1$, para alguna norma matricial, entonces el proceso iterativo

$$x^{(k+1)} = Hx^{(k)} + q,$$

converge hacia la solución del sistema $x = Hx + q$ que existe y es única, para cualquier aproximación inicial $x^{(0)}$.

Demostración: Veamos en primer lugar que el sistema tiene solución única.

El sistema $x = Hx + q$, es decir, $(I - H)x = q$ tiene solución única si y sólo si el sistema homogéneo $(I - H)x = 0$ sólo admite la solución trivial. Si existiese

$z \neq 0$ solución del sistema homogéneo, entonces

$$\|z\| = \|Hz\| \leq \|H\| \|z\| < \|z\|,$$

luego la solución es única. Veamos ahora que la sucesión de iterados $\{x^{(k)}\}$ converge a dicha solución \bar{x} . Para ello,

$$\bar{x} - x^{(k+1)} = (H\bar{x} + q) - (Hx^{(k)} + q) = H(\bar{x} - x^{(k)})$$

y de forma recursiva

$$\bar{x} - x^{(k)} = H^k(\bar{x} - x^{(0)}).$$

Por tanto,

$$\|\bar{x} - x^{(k)}\| \leq \|H^k\| \|\bar{x} - x^{(0)}\| \leq \|H\|^k \|\bar{x} - x^{(0)}\|$$

y como $\|H\| < 1$, se tiene

$$\lim_{k \rightarrow \infty} \|\bar{x} - x^{(k)}\| \leq \lim_{k \rightarrow \infty} \|H\|^k \|\bar{x} - x^{(0)}\| = 0,$$

luego

$$\lim_{k \rightarrow \infty} \|\bar{x} - x^{(k)}\| = 0$$

y

$$\lim_{k \rightarrow \infty} x^{(k)} = \bar{x}.$$

□

Teorema 2

Dado un método iterativo estacionario con matriz de iteración H , las siguientes afirmaciones son equivalentes:

- (a) El método es convergente,
- (b) $\|H\| < 1$,

(c) $\rho(H) < 1$, donde ρ denota el radio espectral,

(d) $\lim_{k \rightarrow \infty} H^k = 0$.

Demostración: Veamos, en primer lugar, $(a) \Rightarrow (d)$.

Si el método es convergente,

$$\lim_{k \rightarrow \infty} e_k = \lim_{k \rightarrow \infty} (x^{(k)} - \bar{x}) = 0,$$

para cualquier $e_0 = x^{(0)} - \bar{x}$. Como

$$e_k = x^{(k)} - \bar{x} = H(x^{(k-1)} - \bar{x}) = \dots = H^k e_0,$$

se debe verificar que $\lim_{k \rightarrow \infty} H^k = 0$. Veamos $(d) \Rightarrow (c)$.

$$\lim_{k \rightarrow \infty} \|H^k v\| \leq \lim_{k \rightarrow \infty} \|H^k\| \|v\| = 0, \quad \forall v \in \mathbb{R}^n,$$

luego

$$\lim_{k \rightarrow \infty} \|H^k v\| = 0, \quad \text{es decir} \quad \lim_{k \rightarrow \infty} H^k v = 0, \forall v \in \mathbb{R}^n.$$

En particular, si tomamos como v el vector propio asociado al valor propio dominante λ_{max} de H , se tiene

$$\lim_{k \rightarrow \infty} H^k v = \lim_{k \rightarrow \infty} \lambda_{max}^k v = 0,$$

luego λ_{max} debe ser menor que la unidad, es decir, $\rho(H) < 1$.

La implicación $(c) \Rightarrow (b)$ se obtiene a partir de la relación que existe entre el radio espectral de una matriz y su norma. Finalmente, La implicación $(b) \Rightarrow (a)$ es evidente a partir del teorema anterior. \square

Método de Jacobi

Sea $A = (a_{ij})$ la matriz del sistema lineal tal que $a_{ii} \neq 0, i = 1, 2, \dots, n$, y consideremos la partición de A de la forma

$$A = L + D + U,$$

donde

- ▶ L es la parte estrictamente triangular inferior de A
- ▶ D es la diagonal principal de A
- ▶ U es la parte estrictamente triangular superior de A

El **Método de Jacobi** es un método estacionario en el que $M = D$ y $N = -(L + U)$, por lo que su expresión iterativa es

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

siendo en este caso $H_J = -D^{-1}(L + U)$ la matriz de iteración.

Lo que básicamente propuso Jacobi es lo siguiente: Dado el sistema $Ax = b$

$$\left. \begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{array} \right\},$$

se transforma en uno equivalente despejando x_1 de la primera ecuación, x_2 de la se-

gunda, y así sucesivamente. De esta forma, resulta

$$\left. \begin{aligned} x_1 &= -\frac{a_{12}}{a_{11}}x_2 - \frac{a_{13}}{a_{11}}x_3 - \cdots - \frac{a_{1n}}{a_{11}}x_n + \frac{b_1}{a_{11}} \\ x_2 &= -\frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 - \cdots - \frac{a_{2n}}{a_{22}}x_n + \frac{b_2}{a_{22}} \\ &\vdots \\ x_n &= -\frac{a_{n1}}{a_{nn}}x_1 - \frac{a_{n2}}{a_{nn}}x_2 - \cdots - \frac{a_{nn-1}}{a_{nn}}x_{n-1} + \frac{b_n}{a_{nn}} \end{aligned} \right\},$$

que coincide con la expresión

$$x = -D^{-1}(L + U)x + D^{-1}b,$$

siendo la expresión iterativa

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

Escalarmente, podemos expresar el proceso iterativo de la forma:

$$\left. \begin{aligned} x_1^{(k+1)} &= -\frac{a_{12}}{a_{11}}x_2^{(k)} - \frac{a_{13}}{a_{11}}x_3^{(k)} - \cdots - \frac{a_{1n}}{a_{11}}x_n^{(k)} + \frac{b_1}{a_{11}} \\ x_2^{(k+1)} &= -\frac{a_{21}}{a_{22}}x_1^{(k)} - \frac{a_{23}}{a_{22}}x_3^{(k)} - \cdots - \frac{a_{2n}}{a_{22}}x_n^{(k)} + \frac{b_2}{a_{22}} \\ &\vdots \\ x_n^{(k+1)} &= -\frac{a_{n1}}{a_{nn}}x_1^{(k)} - \frac{a_{n2}}{a_{nn}}x_2^{(k)} - \cdots - \frac{a_{nn-1}}{a_{nn}}x_{n-1}^{(k)} + \frac{b_n}{a_{nn}} \end{aligned} \right\}.$$

A los resultados de convergencia ya establecidos, podemos añadir el siguiente:

Teorema 3

Supongamos que los elementos diagonales de la matriz $A = (a_{ij})$ son no nulos y que A es estrictamente diagonal dominante, es decir,

$$|a_{ii}| > \sum_{k=1, k \neq i}^n |a_{ik}|, \quad i = 1, 2, \dots, n,$$

entonces el método de Jacobi es convergente para cualquier aproximación inicial $x^{(0)}$.

Demostración: Como ya hemos visto

$$H_J = -D^{-1}(L + U) = I - D^{-1}A = (h_{ij}), \quad \text{con} \quad h_{ii} = 0, \quad h_{ik} = -\frac{a_{ik}}{a_{ii}}, \quad i \neq k.$$

La suma de los elementos de la i -ésima fila de H_J en valor absoluto es

$$\sum_{k=1}^n |h_{ik}| = \frac{1}{|a_{ii}|} \sum_{k=1, k \neq i}^n |a_{ik}|, \quad i = 1, 2, \dots, n,$$

y si la diagonal es estrictamente dominante, $\sum_{k=1}^n |h_{ik}| < 1$, por lo que

$$\|H_J\|_{\infty} = \max_{1 \leq i \leq n} \sum_{k=1}^n |h_{ik}| < 1,$$

por lo que el método es convergente. □

A continuación describimos en forma de algoritmo el método de Jacobi.

ALGORITMO

Entrada: Matriz A , términos independientes b , aproximación inicial $x^{(0)}$, tolerancia Tol , número máximo de iteraciones $maxiter$

Salida: Solución aproximada o mensaje de fracaso

Paso 1 Tomar $iter = 1$, $incre = tol + 1$

Paso 2 Calcular las matrices D , L y U

Paso 3 Mientras $iter \leq maxiter$ y $incre > tol$

Paso 4 Tomar $x = -D^{-1}(L + U)x^{(0)} + D^{-1}b$

Paso 5 $incre = \|x - x^{(0)}\|$

Paso 6 $iter = iter + 1$

Paso 7 $x^{(0)} = x$

Paso 8 Analizar por qué nos hemos salido del bucle

Es un algoritmo sencillo de implementar en Matlab, tal como podemos observar en el siguiente fichero.



```
function [x,iter,incre]=Jacobi(A,b,x0,tol,maxiter)
%Metodo de Jacobi para resolver Ax=b

L=tril(A,-1);
U=triu(A,1);
d=diag(A);
iD=diag(1./d);
iter=1;
incre=tol+1;
v=[];
while iter<=maxiter && incre>tol
    x = -iD*(L+U)*x0+iD*b;
    incre = norm(x-x0,inf)/norm(x,inf);
    v=[v incre];
    iter = iter+1;
    x0=x;
end
if iter>maxiter
    disp('se ha alcanzado el maximo de iteraciones')
end

end
```

Ejemplo 1.

Consideremos el sistema de tamaño 4×4

$$\left. \begin{array}{rrcr} 10x_1 & -x_2 & +2x_3 & & = & 6 \\ -x_1 & +11x_2 & -x_3 & +3x_4 & = & 25 \\ 2x_1 & -x_2 & +10x_3 & -x_4 & = & -11 \\ & 3x_2 & -x_3 & +8x_4 & = & 15 \end{array} \right\}$$

Aplicando el método de Jacobi, con una tolerancia de 10^{-3} y el criterio de parada

$$\frac{\|x^{(k+1)} - x^{(k)}\|_{\infty}}{\|x^{(k+1)}\|_{\infty}} < 10^{-3},$$

se obtienen los resultados que aparecen en la Tabla 1 correspondientes a las 10 primeras iteraciones, valor del contador en el que ha parado el programa.

k	0	1	2	3	...	9	10
$x_1^{(k)}$	0.0000	0.0600	1.0473	0.9326		0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.0533		2.0004	1.9998
$x_3^{(k)}$	0.0000	-1.1000	-0.8052	-1.0493		-1.0004	-0.9999
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309		1.0006	0.9999

Tabla 1: Valores de las 10 primeras iteraciones

Teniendo en cuenta que la solución exacta es $\bar{x} = (1, 2, -1, 1)^T$, obtenemos como error exacto en la última iteración

$$\|x^{(10)} - \bar{x}\|_{\infty} = 0.0002.$$

Método de Gauss-Seidel

Este método pretende aumentar la velocidad de convergencia del método de Jacobi. Partiendo de su misma idea, se sustituye a la derecha de la expresión de punto fijo las componentes del iterado k o del $k + 1$ si ya han sido calculadas. De esta forma, la expresión iterativa escalar de este método resulta

$$\left. \begin{aligned} x_1^{(k+1)} &= -\frac{a_{12}}{a_{11}}x_2^{(k)} - \frac{a_{13}}{a_{11}}x_3^{(k)} - \dots - \frac{a_{1n}}{a_{11}}x_n^{(k)} + \frac{b_1}{a_{11}} \\ x_2^{(k+1)} &= -\frac{a_{21}}{a_{22}}x_1^{(k+1)} - \frac{a_{23}}{a_{22}}x_3^{(k)} - \dots - \frac{a_{2n}}{a_{22}}x_n^{(k)} + \frac{b_2}{a_{22}} \\ &\vdots \\ x_n^{(k+1)} &= -\frac{a_{n1}}{a_{nn}}x_1^{(k+1)} - \frac{a_{n2}}{a_{nn}}x_2^{(k+1)} - \dots - \frac{a_{nn-1}}{a_{nn}}x_{n-1}^{(k+1)} + \frac{b_n}{a_{nn}} \end{aligned} \right\}.$$

Volviendo a utilizar la partición de la matriz A del sistema en $A = L + D + U$ y tomando $M = D + L$ y $N = -U$, obtenemos la expresión matricial del **Método de Gauss-Seidel** cuya expresión iterativa es

$$x^{(k+1)} = -(D + L)^{-1}Ux^{(k)} + (D + L)^{-1}b, \quad k = 0, 1, \dots$$

Observemos que $(D + L)^{-1}$ ya no es tan inmediata de calcular como D^{-1} que aparecía en el método de Jacobi y su cálculo debe ser evitado. Por ello:

- En cada iteración debemos resolver el sistema $(D + L)x^{(k+1)} = b - Ux^{(k)}$
- En el método de Gauss-Seidel las componentes de $x^{(k+1)}$ que ya conocemos se utilizan en la propia iteración $k + 1$

Expresamos este método en forma de algoritmo para su sencilla implementación.

ALGORITMO

Entrada: Matriz A , términos independientes b , aproximación inicial $x^{(0)}$, tolerancia Tol , número máximo de iteraciones *maxiter*

Salida: Solución aproximada o mensaje de fracaso

Paso 1 Tomar $iter = 1$ y $incre = tol + 1$

Paso 2 Obtener las matrices D , L y U

Paso 3 Mientras $iter \leq maxiter$ y $incre > tol$

Paso 4 Obtener x solución del sistema $(D + L)x = b - Ux^{(0)}$

Paso 5 Actualizar $incre = \|x - x^{(0)}\|$

Paso 6 Actualizar $iter = iter + 1$

Paso 7 $x^{(0)} = x$

Paso 8 Analizar por qué nos hemos salido del bucle.

Para el paso 4, debemos observar que la matriz de coeficientes es triangular inferior

$$\begin{pmatrix} c_{11} & 0 & 0 & \cdots & 0 \\ c_{21} & c_{22} & 0 & \cdots & 0 \\ c_{31} & c_{32} & c_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{pmatrix},$$

por lo que debemos utilizar el procedimiento de sustitución directa para resolver el sistema

$$\begin{aligned} x_1 &= d_1/c_{11}, \\ x_2 &= d_2/c_{22} - c_{21}x_1/c_{22}, \\ x_3 &= d_3/c_{33} - c_{32}x_2/c_{33} - c_{31}x_1/c_{33}, \\ &\vdots \\ x_n &= d_n/c_{nn} - c_{nn-1}x_{n-1}/c_{nn} - c_{nn-2}x_{n-2}/c_{nn} - \cdots - c_{n1}x_1/c_{nn}. \end{aligned}$$

Con todo ello, el fichero .m en el que implementamos el algoritmo de Gauss-Seidel queda de la siguiente forma.



GaussSeidel.m

```
function [Sol,iter,incre] = ...  
    GaussSeidel(A,b,x0,maxiter,tol)  
% Algoritmo de Gauss-Seidel para sistemas Ax=b  
b=b(:);  
x0=x0(:);  
D=diag(diag(A));  
L=tril(A,-1);  
U=triu(A,1);  
x=x0;  
iter=1;  
incre=tol+1;  
while iter<maxiter && incre >tol  
    B=L+D;  
    d=b-U*x0;  
    x(1)=d(1)/B(1,1);  
    for j=2:length(d)  
        x(j)=(d(j)-B(j,1:j-1)*x(1:j-1))/B(j,j);  
    end  
    incre=norm(x0-x);  
    iter=iter+1;  
    x0=x;  
end  
if incre>tol  
    disp('necesito mas iteraciones');  
else  
    Sol=x;  
end  
end
```

Ejemplo 2.

Consideremos el sistema de tamaño 4×4 que ya utilizamos en el ejemplo anterior

$$\left. \begin{array}{rrcr} 10x_1 & -x_2 & +2x_3 & & = & 6 \\ -x_1 & +11x_2 & -x_3 & +3x_4 & = & 25 \\ 2x_1 & -x_2 & +10x_3 & -x_4 & = & -11 \\ & 3x_2 & -x_3 & +8x_4 & = & 15 \end{array} \right\}$$

Aplicando el método de Gauss-Seidel, con una tolerancia de 10^{-3} y el criterio de parada

$$\frac{\|x^{(k+1)} - x^{(k)}\|_{\infty}}{\|x^{(k+1)}\|_{\infty}} < 10^{-3},$$

se obtienen los resultados que aparecen en la Tabla 2 correspondientes a las 5 primeras iteraciones.

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.0600	1.0300	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.0370	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	-0.9873	-1.0140	-1.0025	-1.0003	-1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Tabla 2: Valores de las 5 primeras iteraciones

Teniendo en cuenta que la solución exacta es $\bar{x} = (1, 2, -1, 1)^T$, obtenemos

$$\|x^{(5)} - \bar{x}\|_{\infty} = 0.00002.$$

Es interesante comparar los resultados obtenidos, para este sistema, por los métodos de Jacobi (ver Ejemplo 1) y de Gauss-Seidel (ver Ejemplo 2). Es fácil apreciar una aceleración de la convergencia en un método frente al otro.

La matriz de iteración en el caso del método de Gauss-Seidel es $H_{GS} = -(D+L)^{-1}U$. Al igual que en el caso de Jacobi, podemos establecer un nuevo resultado de conver-

gencia con condiciones sobre la matriz del sistema.

Teorema 4

Supongamos que los elementos diagonales de la matriz $A = (a_{ij})$ son no nulos y que A es estrictamente diagonal dominante, entonces el método de Gauss-Seidel es convergente para cualquier aproximación inicial $x^{(0)}$.

Demostración: Vamos a demostrar que $\|H_{GS}\|_{\infty} < 1$, para lo que recordamos que

$$\|H_{GS}\|_{\infty} = \sup_{\|x\|_{\infty}=1} \|H_{GS}x\|.$$

Sea $x \in \mathbb{R}^n$, $\|x\|_{\infty} = 1$, y analicemos las componentes del vector $y = H_{GS}x$. Podemos interpretar y como el vector resultado de una etapa de Gauss-Seidel sobre x , en un sistema con $b = 0$. Así,

$$|y_1| = |(-a_{12}x_2 - \dots - a_{1n}x_n)/a_{11}| \leq \|x\|_{\infty} (|a_{12}| + \dots + |a_{1n}|) / |a_{11}| < 1.$$

$$\begin{aligned} |y_2| &= |(-a_{21}y_1 - a_{23}x_3 - \dots - a_{2n}x_n)/a_{22}| \\ &\leq \sup(|y_1|, |x_3|, \dots, |x_n|) \left(\sum_{k=1, k \neq 2}^n |a_{2k}| \right) / |a_{22}| < 1. \end{aligned}$$

...

$$|y_n| = \sup(|y_1|, |y_2|, \dots, |y_{n-1}|) \left(\sum_{k=1}^{n-1} |a_{nk}| \right) / |a_{nn}| < 1.$$

Por tanto, $\|H_{GS}\|_{\infty} < 1$. □

También para este método se satisface el siguiente resultado:

Teorema 5

Si $\|H_{GS}\| < 1$ para cualquier norma matricial, entonces la sucesión $\{x^{(k)}\}$ obtenida por el método de Gauss-Seidel converge a la solución del sistema, para

cualquier $x^{(0)}$ inicial, y se satisfacen las cotas de error:

$$\|\bar{x} - x^{(k)}\| \leq \|H\|^k \|\bar{x} - x^{(0)}\|$$

y

$$\|\bar{x} - x^{(k)}\| \leq \frac{\|H\|^k}{1 - \|H\|} \|x^{(1)} - x^{(0)}\|.$$

Métodos iterativos SOR

En los métodos de relajación interviene un parámetro w del que depende, en gran medida, la convergencia de dichos métodos. Se basan en la descomposición de A de la forma

$$A = \frac{1}{w}D - \frac{1-w}{w}D + L + U = \frac{1}{w}(D + wL) - \left(\frac{1-w}{w}D - U\right).$$

De aquí,

$$Ax = b \quad \Leftrightarrow \quad \frac{1}{w}(D + wL)x = \left(\frac{1-w}{w}D - U\right)x + b \quad \Leftrightarrow$$

$$(D + wL)x = ((1-w)D - wU)x + wb.$$

De esta expresión podemos obtener los métodos iterativos

► Método SOR1 (successive over relaxation)

$$(D + wL)x^{(k+1)} = ((1-w)D - wU)x^{(k)} + wb.$$

► Método SOR2

$$(D + wU)x^{(k+1)} = ((1-w)D - wL)x^{(k)} + wb.$$

La expresión iterativa ecuación a ecuación, resulta

$$\left. \begin{aligned} x_1^{(k+1)} &= (1-w)x_1^{(k)} + w \left(-\frac{a_{12}}{a_{11}}x_2^{(k)} - \frac{a_{13}}{a_{11}}x_3^{(k)} - \dots - \frac{a_{1n}}{a_{11}}x_n^{(k)} + \frac{b_1}{a_{11}} \right) \\ x_2^{(k+1)} &= (1-w)x_2^{(k)} + w \left(-\frac{a_{21}}{a_{22}}x_1^{(k+1)} - \frac{a_{23}}{a_{22}}x_3^{(k)} - \dots - \frac{a_{2n}}{a_{22}}x_n^{(k)} + \frac{b_2}{a_{22}} \right) \\ &\vdots \\ x_n^{(k+1)} &= (1-w)x_n^{(k)} + w \left(-\frac{a_{n1}}{a_{nn}}x_1^{(k+1)} - \frac{a_{n2}}{a_{nn}}x_2^{(k+1)} - \dots - \frac{a_{nn-1}}{a_{nn}}x_{n-1}^{(k+1)} + \frac{b_n}{a_{nn}} \right) \end{aligned} \right\}.$$

Si $\bar{x}^{(k+1)}$ denota el iterado $k+1$ del método de Gauss-Seidel, la expresión iterativa vectorial del método SOR1 se puede escribir de la forma:

$$x^{(k+1)} = (1-w)x^{(k)} + w\bar{x}^{(k+1)}, \quad k = 0, 1, 2, \dots,$$

siendo en este caso la matriz de iteración $H_w = (D + wL)^{-1}((1-w)D - wU)$. Esto nos permite establecer el siguiente resultado que nos ayudará en el análisis de la convergencia del método SOR1.

Teorema 6

La matriz de iteración H_w del método SOR1 verifica que $\rho(H_w) \geq |w-1|$.

Demostración: A partir de la expresión de H_w ,

$$H_w = (D + wL)^{-1}DD^{-1}((1-w)D - wU) = (I + wD^{-1}L)^{-1}((1-w)I - wD^{-1}U).$$

Calculemos el polinomio característico de H_w

$$\begin{aligned} p(\lambda) &= \det(\lambda I - H_w) = \det(\lambda I - (I + wD^{-1}L)^{-1}((1-w)I - wD^{-1}U)) \\ &= \det(\lambda(I + wD^{-1}L)^{-1}(I + wD^{-1}L) \\ &\quad - (I + wD^{-1}L)^{-1}((1-w)I - wD^{-1}U)) \\ &= \det(I + wD^{-1}L)^{-1} \det(\lambda(I + wD^{-1}L) - ((1-w)I - wD^{-1}U)). \end{aligned}$$

La matriz $I + wD^{-1}L$ es triangular inferior con unos en la diagonal, por lo que $(I + wD^{-1}L)^{-1}$ también será triangular inferior con unos en la diagonal, luego

$\det(I + wD^{-1}L)^{-1} = 1$. Por tanto,

$$p(\lambda) = \det(\lambda(I + wD^{-1}L) - ((1 - w)I - wD^{-1}U)).$$

Si $\lambda_1, \lambda_2, \dots, \lambda_n$ son los valores propios de H_w , resulta

$$\rho(H_w)^n \geq \prod_{i=1}^n |\lambda_i|$$

y además $|p(0)| = \prod_{i=1}^n |\lambda_i|$. Por tanto,

$$\rho(H_w)^n \geq |p(0)| = |\det((w - 1)I + wD^{-1}U)|.$$

La matriz $D^{-1}U$ es estrictamente triangular superior, por lo que $((w - 1)I + wD^{-1}U)$ es triangular superior con elementos de la diagonal iguales a $w - 1$. Así pues,

$$\rho(H_w)^n \geq |w - 1|^n, \quad \Rightarrow \quad \rho(H_w) \geq |w - 1|.$$

□

A partir de este resultado podemos concluir:

Teorema 7

Una condición necesaria para que el método SOR1 converja es que $w \in]0, 2[$.

A continuación, señalamos algunos hechos sobre la convergencia del método SOR1:

- La condición $w \in]0, 2[$ es sólo necesaria. Hay tipos de matrices para las cuales es condición necesaria y suficiente.
- ¿Cuál es el valor óptimo de w ? La respuesta depende de las características de la matriz A .

- Teniendo en cuenta que el método SOR1 puede escribirse de la forma

$$x^{(k+1)} = x^{(k)} - wD^{-1}(Lx^{(k+1)} - Dx^{(k)} + Ux^{(k)} + b),$$

el **factor de proporcionalidad** que multiplica el error es w/a_{ii} mientras que en método de Gauss-Seidel era $1/a_{ii}$.

- Si A es una matriz tridiagonal y $\rho(H_J)$ el radio espectral de la matriz de Jacobi, el parámetro óptimo de relajación viene dado por

$$w_{op} = \frac{2}{1 + (1 - (\rho(H_J))^2)^{1/2}}.$$

Teorema 8

Si A es una matriz diagonal dominante, entonces el método SOR converge para cualquier $x^{(0)}$ y cualquier valor de w tal que $0 < w \leq 1$.

En el siguiente ejemplo, vamos a comparar los resultados obtenidos con el método de Gauss-Seidel y el método SOR1.

Ejemplo 3.

Consideremos el sistema

$$\left. \begin{array}{rcl} 4x_1 + 3x_2 & = & 24 \\ 3x_1 + 4x_2 - x_3 & = & 30 \\ -x_2 + 4x_3 & = & -24 \end{array} \right\}$$

que tiene como solución exacta $\bar{x} = (3, 4, -5)^T$. Aplicando el método de Gauss-Seidel y el criterio de parada

$$\|x^{(k+1)} - x^{(k)}\|_{\infty} < 10^{-7},$$

se obtienen los resultados

k	0	1	2	3	4	5	6
$x_1^{(k)}$	1.0000	5.2500	3.1406	3.0878	3.0549	3.0343	3.0214
$x_2^{(k)}$	1.0000	3.8125	3.8828	3.9267	3.9542	3.9713	3.9821
$x_3^{(k)}$	1.0000	-5.0468	-5.0292	-5.0183	-5.0114	-5.0071	-5.0044

Tabla 3: Valores de las 6 primeras iteraciones en el método de Gauss-Seidel

Para que se cumpla el criterio de parada necesitamos 34 iteraciones.

Vamos a aplicar ahora el método SOR1 con $w = 1.25$ y el mismo criterio de parada

$$\|x^{(k+1)} - x^{(k)}\|_{\infty} < 10^{-7}.$$

Los resultados obtenidos se muestran en la siguiente tabla.

k	0	1	2	3	4	5	6
$x_1^{(k)}$	1.0000	6.3125	2.6223	3.1330	2.9570	3.0037	2.9963
$x_2^{(k)}$	1.0000	3.5195	3.9585	4.0102	4.0074	4.0029	4.0009
$x_3^{(k)}$	1.0000	-6.6501	-4.6004	-5.0966	-4.9734	-5.0037	-4.9982

Tabla 4: Valores de las 6 primeras iteraciones con el método SOR1

Con el método SOR1, para que se cumpla el criterio de parada necesitamos 14 iteraciones.



Accede al vídeo: Aplicación del método de Jacobi para resolver un problema de contorno

8.3 Problemas de contorno elípticos

Como ya mencionamos al principio del tema, la descripción más sencilla de un problema de este tipo es

$$\nabla^2 u(x, y) \equiv u_{xx}(x, y) + u_{yy}(x, y) = f(x, y), \quad (x, y) \in R,$$

$$u(x, y) = g(x, y) \text{ para } (x, y) \in S,$$

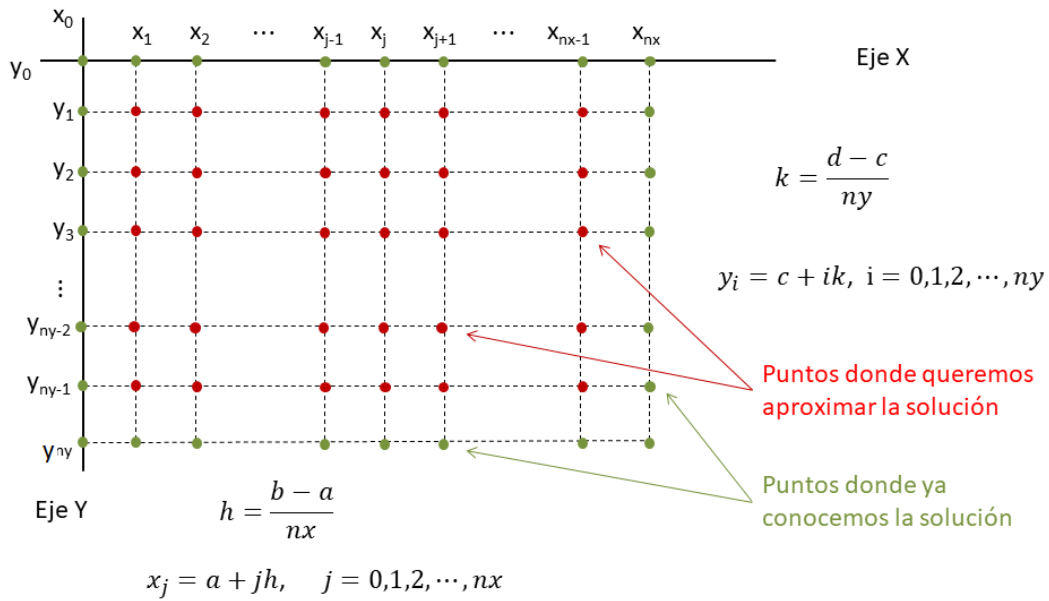
donde $R = \{(x, y) : a < x < b, c < y < d\}$ y S es la frontera de R .

Mediante diferencias finitas transformamos el problema en un sistema de ecuaciones lineales (ya que la EDP es lineal) cuya solución nos dará valores aproximados del problema de contorno en los puntos elegidos. Las condiciones de frontera pueden ser tipo Dirichlet, naturales o mixtas.

Establecemos el mallado de los puntos donde vamos a aproximar la solución del problema

$$(x, y) \in]a, b[\times]c, d[,$$

en el supuesto de tener condiciones Dirichlet. Observemos en la siguiente gráfica los puntos en los que la solución es conocida y en los que debemos aproximarla.



Tomamos nx subintervalos para la variable x y ny para la variable y .

$$h = \frac{b-a}{nx}, k = \frac{d-c}{ny}, x_j = a + jh, j = 0, 1, \dots, nx; y_i = c + ik, i = 0, 1, \dots, ny.$$

Aproximamos las derivadas parciales que aparecen en la EDP mediante diferencias finitas centrales

$$u_{xx}(y_i, x_j) \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}, u_{yy}(y_i, x_j) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{k^2}.$$

Sustituyendo en la ecuación, obtenemos

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{k^2} = f(y_i, x_j)$$

y llamando $\lambda = \frac{h}{k}$, resulta

$$2(\lambda^2 + 1)u_{i,j} - (u_{i,j+1} + u_{i,j-1}) - \lambda^2(u_{i+1,j} + u_{i-1,j}) = -h^2 f(y_i, x_j),$$

con $j = 1, 2, \dots, nx - 1$ y $i = 1, 2, \dots, ny - 1$ y las condiciones de contorno

$$u_{j,0} = g(a, y_j), \quad u_{j,nx} = g(b, y_j), \quad j = 0, 1, \dots, nx,$$

$$u_{0,i} = g(x_i, c), \quad u_{ny,i} = g(x_i, d), \quad i = 0, 1, \dots, ny.$$

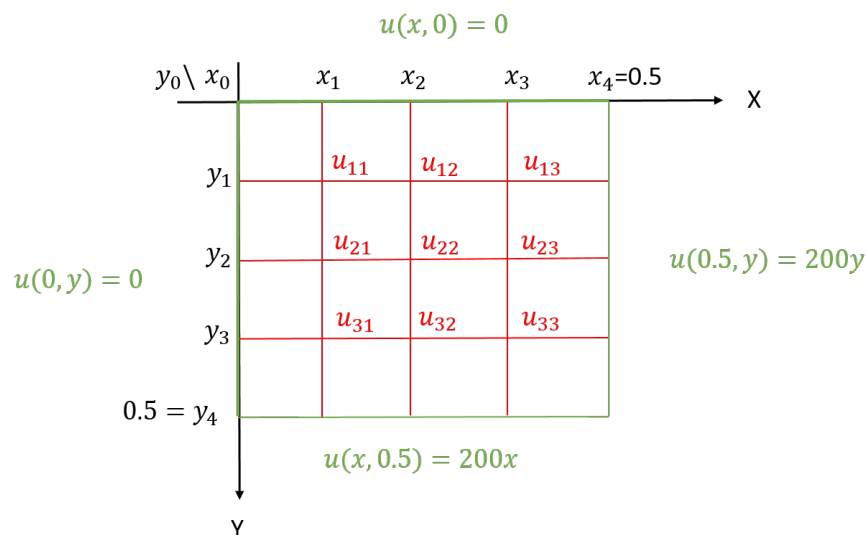
Ejemplo 4.

Consideremos el problema elíptico

$$u_{xx}(x, y) + u_{yy}(x, y) = 0, \quad (x, y) \in R = [0, 0.5] \times [0, 0.5],$$

$$u(x, 0) = 0, \quad u(x, 0.5) = 200x \quad u(0, y) = 0, \quad u(0.5, y) = 200y.$$

Tomamos 4 subintervalos en cada eje, $nx = 4$ y $ny = 4$. Observamos en la gráfica el mallado que hemos generado.



Al reemplazar las derivadas parciales por diferencias finitas centrales y sustituir en cada punto (x_i, y_j) del mallado, obtenemos el sistema

$$4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} = 0, \quad i, j = 1, 2, 3,$$

que matricialmente resulta

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{1,2} \\ u_{1,3} \\ u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{3,1} \\ u_{3,2} \\ u_{3,3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 25 \\ 0 \\ 0 \\ 50 \\ 25 \\ 50 \\ 125 \end{pmatrix}.$$

Quizás no en este ejemplo, debido a que n_x y n_y son valores pequeños, pero podría ser complicado introducir por teclado la matriz de coeficientes y los términos independientes, para aplicar los métodos descritos en la sección anterior. Es conveniente buscar una estructura en la matriz a la hora de introducirla en el ordenador. Por ejemplo, si llamamos

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}, I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, O_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

U al vector de incógnitas (9 componentes) y d a los términos independientes (9 componentes), entonces el sistema anterior se puede escribir en la forma

$$\begin{pmatrix} A & -I_3 & O_3 \\ -I_3 & A & -I_3 \\ O_3 & -I_3 & A \end{pmatrix} \begin{pmatrix} U \end{pmatrix} = \begin{pmatrix} d \end{pmatrix}.$$

También podríamos introducir la matriz del sistema mediante diagonales.

En cualquier caso, el algoritmo para resolver el problema elíptico queda de la siguiente forma, en la que no hemos establecido el método iterativo utilizado en la resolución del sistema.

ALGORITMO

- ▶ Entrada: funciones f, g ; valores a, b, c, d ; enteros nx, ny ; tolerancia tol ; número máximo de iteraciones $maxiter$.
- ▶ Salida: aproximaciones $u_{i,j}$ de $u(x_i, y_j)$, $i = 1, 2, \dots, nx-1$, $j = 1, 2, \dots, ny-1$, o un mensaje de fracaso.
- ▶ Paso 1. Tomar $h = (b - a)/nx$, $k = (d - c)/ny$.
- ▶ Paso 2. Elegir los nodos del método: $x = a : h : b$, $y = c : k : d$
- ▶ Paso 3. Inicializar con ceros la matriz U de tamaño $ny + 1 \times nx + 1$
- ▶ Paso 4. Rellenar con las condiciones de contorno las filas y columnas correspondientes de U
- ▶ Paso 5. Tomar $\lambda = h/k$, $iter = 1$ y $incre = tol + 1$, $U^{(0)} = zeros$
- ▶ Paso 6. Mientras $iter \leq maxiter$ y $incre > tol$
 - Paso 7. Método iterativo para resolver el sistema lineal $\rightarrow U$
 - Paso 8. $incre = \max(\max(U - U^{(0)}))$
 - Paso 9. $U^{(0)} = U$, $iter = iter + 1$
- ▶ Paso 10. Analizar porqué el ordenador se ha salido del bucle.

En el sistema lineal obtenido al aplicar diferencias finitas al problema elíptico, podemos evitar tener que introducir la matriz de coeficientes y los términos independientes al ordenador si procedemos de la siguiente forma:

Escribimos el sistema a resolver de la forma:

$$u_{i,j} = \frac{1}{2(\lambda^2 + 1)} [\lambda^2(u_{i+1,j} + u_{i-1,j}) + u_{i,j+1} + u_{i,j-1} - h^2 f(y_i, x_j)],$$

donde $j = 1, 2, \dots, nx-1$, $i = 1, 2, \dots, ny-1$. Podemos aplicar la idea del método de Jacobi, interpretándolo como un método de punto fijo que, al sustituir un iterado

en la parte derecha de las ecuaciones y haciendo operaciones se obtiene el nuevo iterado. Así

METODO DE JACOBI

Para $j = 2 : nx$

Para $i = 2 : ny$

$$u_{i,j}^{(k+1)} = \frac{1}{2(\lambda^2 + 1)} \left[\lambda^2 (u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)}) + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)} - h^2 f(y_i, x_j) \right],$$

Fin para i

Fin para j

Podemos utilizar como criterio de parada

$$\max_i \left(\max_j \left(\left| u_{i,j}^{(k+1)} - u_{i,j}^{(k)} \right| \right) \right) < tol.$$

De forma análoga, los otros dos métodos iterativos estudiados resultan:

METODO DE GAUSS-SEIDEL

Para $j = 2 : nx$

Para $i = 2 : ny$

$$u_{i,j}^{(k+1)} = \frac{1}{2(\lambda^2 + 1)} \left[\lambda^2 (u_{i+1,j}^{(k)} + u_{i-1,j}^{(k+1)}) + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k+1)} - h^2 f(y_i, x_j) \right],$$

Fin para i

Fin para j

METODO SOR1

Para $j = 2 : nx$

Para $i = 2 : ny$

$$\bar{u}_{i,j}^{(k+1)} \frac{1}{2(\lambda^2+1)} \left[\lambda^2 (u_{i+1,j}^{(k)} + u_{i-1,j}^{(k+1)}) + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k+1)} - h^2 f(y_i, x_j) \right],$$

$$u_{i,j}^{(k+1)} = (1 - w)u_{i,j}^{(k)} + w\bar{u}_{i,j}^{(k+1)}$$

Fin para i

Fin para j

A continuación mostramos el archivo de función de Matlab en el que hemos implementado el método de Gauss-Seidel para resolver el sistema correspondiente al problema elíptico.



```
function ...  
    [U, incre, iter, x, y] = ElipticaGaussSeidel(f0, f1, g0, g1, ...  
        a, b, c, d, nx, ny, funcionti, maxiter, tol)  
% Metodo Gauss-Seidel para aproximar la solucion ...  
% del problema eliptico  
h = (b - a) / nx;  
x = a:h:b;  
k = (d - c) / ny;  
y = c:k:d;  
f0x = feval(f0, x);    f1x = feval(f1, x);  
g0y = feval(g0, y);    g1y = feval(g1, y);  
u0 = (sum(f0x + f1x) / (nx + 1) + sum(g0y + g1y) / (ny + 1)) / 4;  
U = u0 * ones(ny + 1, nx + 1);  
  
U(1, :) = feval(f0, x);  
U(ny + 1, :) = feval(f1, x);  
U(:, 1) = feval(g0, y);  
U(:, nx + 1) = feval(g1, y);
```

```

iter=1;
incre=tol+1;
lambda=h/k;
p=1/(2*(lambda^2+1));
q=lambda^2*p;
r=-h^2*p;
V=U;
while iter<maxiter && incre>tol
    for j=2:nx
        for i=2:ny
            V(i,j)=q*(U(i+1,j)+V(i-1,j))+...
                p*(U(i,j+1)+V(i,j-1))+...
                r*feval(funcionti,x(j),y(i));
        end
    end
    incre= max(max(abs(U-V)));
    U=V;
    iter=iter+1;
end
end

```

Ejemplo 5.

Consideremos el problema elíptico modelizado con la siguiente ecuación de Poisson y las condiciones de contorno Dirichlet

$$\begin{aligned}
 u_{xx}(x,y) + u_{yy}(x,y) &= xe^y, \quad (x,y) \in R =]0,2[\times]0,1[, \\
 u(x,0) &= x, \quad u(x,1) = xe, \quad u(0,y) = 0, \quad u(2,y) = 2e^y.
 \end{aligned}$$

Sabemos que la solución exacta del problema es $u(x,y) = xe^y$. Mediante diferencias finitas transformamos el problema elíptico en un sistema lineal, utilizando $nx = 6$ y $ny = 5$. Aplicamos el método de Gauss-Seidel para resolver dicho siste-

ma lineal, usando como criterio de parada

$$\left| u_{i,j}^{(k)} - u_{i,j}^{(k-1)} \right| \leq 10^{-10}.$$

Mostramos en la columna 5 de la Tabla 5 los resultados obtenidos.

i	j	x_j	y_i	$u_{i,j}$	$u(x_j, y_i)$	$ u_{i,j} - u(x_j, y_i) $
1	1	0.3333	0.2	0.40726	0.40713	1.3e-4
1	2	0.3333	0.4	0.49748	0.49727	2.08e-4
1	3	0.3333	0.6	0.60760	0.60737	2.23e-4
1	4	0.3333	0.8	0.74201	0.74185	1.6e-4
2	1	0.6667	0.2	0.81452	0.81427	2.5e-4
2	2	0.6667	0.4	0.99496	0.99455	4.08e-4
2	3	0.6667	0.6	1.21520	1.21470	4.37e-4
2	4	0.6667	0.8	1.48400	1.48370	3.15e-4
3	1	1.0	0.2	1.22180	1.22140	3.64e-4
3	2	1.0	0.4	1.49240	1.49180	5.8e-4
3	3	1.0	0.6	1.82270	1.82210	6.24e-4
⋮	⋮					
5	3	1.6667	0.6	3.03750	3.03690	6.4e-4
5	4	1.6667	0.8	3.70970	3.70920	4.8e-4

Tabla 5: Valores aproximados, exactos y error en cada punto

En la columna 6 de la Tabla 5 mostramos el valor exacto en cada punto del malla-
do, mientras que en la última columna aparece el error exacto calculado.

En el siguiente ejemplo, aproximamos la solución de un problema elíptico en el que
alguna de sus condiciones de contorno no es Dirichlet.

Ejemplo 6.

Consideremos la ecuación de Laplace con condiciones de contorno:

$$\begin{aligned} u_{xx} + u_{yy} &= 0, & -1 < x < 1, \quad -1 < y < 1, \\ u(x, 1) &= 0, & u(x, -1) &= 1, \\ u_x(1, y) &= u_x(-1, y), & u_x(-1, y) &= -\frac{1}{2}u(-1, y). \end{aligned} \quad (1)$$

- Transforma este problema en un sistema lineal, aplicando diferencias finitas de segundo orden con paso $h = k = 0.25$ para x e y , respectivamente.
- Aplica el método de Gauss-Seidel con tolerancia $tol = 10^{-5}$ para resolver el sistema anterior. ¿Cuántas iteraciones son necesarias?

Definimos, en primer lugar, sendas particiones de los intervalos en que están definidas las variables x e y :

$$\begin{aligned} x_i &= -1 + ih, & \text{con } h &= 2/nx, \quad i = 0, 1, \dots, nx, \\ y_j &= -1 + jk, & \text{con } k &= 2/ny, \quad j = 0, 1, \dots, ny. \end{aligned}$$

A continuación, discretizamos la ecuación en derivadas parciales mediante diferencias finitas centrales.

$$\begin{aligned} u_{i,j} &= \frac{1}{2(1 + \lambda^2)} \left(\lambda^2(u_{i+1,j} + u_{i-1,j}) + u_{i,j-1} + u_{i,j+1} \right), \\ i &= 0, 1, \dots, nx, \quad j = 1, 2, \dots, ny - 1, \end{aligned}$$

donde $\lambda = k/h$.

Observemos que para $i = 0$ e $i = nx$ aparece la función incógnita evaluada en el nodo x_{-1} , y x_{nx+1} , respectivamente. Estos nodos, evidentemente, no pertenecen al intervalo $[-1, 1]$, por lo que utilizaremos las condiciones que sobre la derivada de la función incógnita nos proporciona el enunciado del problema para

aproximar el valor de la función incógnita en estos nodos.

$$u_x(-1, y_j) = \frac{u(-1+h, y_j) - u(-1-h, y_j)}{2h} + O(h^2) = -\frac{1}{2}u(-1, y_j),$$

$$u_x(1, y_j) = \frac{u(1+h, y_j) - u(1-h, y_j)}{2h} + O(h^2) = u(1, y_j),$$

de donde obtenemos

$$u_{-1,j} = hu_{0,j} + u_{1,j},$$

$$u_{nx+1,j} = 2hu_{nx,j} + u_{nx-1,j}.$$

El sistema resultante queda:

$$u_{0,j} = \frac{2\lambda^2 u_{1,j} + u_{0,j-1} + u_{0,j+1}}{2(1+\lambda^2) - h\lambda^2}, \quad j = 1, \dots, ny-1,$$

$$u_{i,j} = \frac{\lambda^2(u_{i+1,j} + u_{i-1,j}) + u_{i,j-1} + u_{i,j+1}}{2(1+\lambda^2)}, \quad \begin{matrix} i = 1, \dots, nx-1, \\ j = 1, \dots, ny-1, \end{matrix}$$

$$u_{nx,j} = \frac{2\lambda^2 u_{nx-1,j} + u_{nx,j+1} + u_{nx,j-1}}{2(1+\lambda^2) - 2h\lambda^2}, \quad j = 1, \dots, ny-1.$$

Para aplicar el método de Gauss-Seidel al sistema lineal, generamos la función GSEDP1.m. Hay ciertas variaciones con respecto al problema de Laplace estándar.

- Inicialización de la matriz U . No conocemos el valor de la función incógnita para $x = -1$ y $x = 1$. Así, inicializamos la primera fila $u(-1, y_j)$, $j = 0, 1, \dots, ny$ como un vector progresivo desde $u(-1, -1) = 1$ hasta $u(-1, 1) = 0$. Del mismo modo, inicializamos la última fila $u(1, y_j)$, $j = 0, 1, \dots, ny$ como el vector progresivo definido desde $u(1, -1) = 1$ hasta $u(1, 1) = 0$.
- Inicializamos el resto de la matriz como el promedio de los valores de contorno.
- Dos bucles anidados que nos permitan movernos por todos los pares de nodos centrales. En el interior del doble bucle, para cada valor fijo de los índices i y j , recalculamos por separado $u_{0,j}$, $u_{i,j}$ y $u_{nx,j}$ empleando las expresiones

anteriores.

En el siguiente archivo .m hemos adaptado nuestro programa de Gauss-Seidel a este ejemplo concreto.



```
function [U,iter] = ...
    GSEDP1(a,b,c,d,h,k,f0,f1,tol,maxiter)

x = a:h:b;    nx = (b-a)/h;    ny = (d-c)/k;
lambda = k/h;    lambda2 = lambda^2;
beta = 2*(1+lambda2);    beta2 = beta-2*h*lambda2;
f0x = feval(f0,x);    f1x = feval(f1,x);

g0 = f0x(1):-k/2:f1x(1);    g1 = g0;
u0 = (sum(f0x+f1x)/(nx+1)+sum(g0+g1)/(ny+1))/4; ...
    % inicializacion
U = u0*ones(nx+1,ny+1);
U(1,:) = g0;    U(nx+1,:) = g1;
U(:,1) = f0x(:);    U(:,ny+1) = f1x(:);

incr = tol+1;    iter = 0;
while incr>tol && iter<maxiter
    incr = 0;
    for i = 2:nx
        for j = 2:ny
            U(1,j) = (2*lambda2*U(2,j) + ...
                U(1,j-1) + U(1,j+1)) ...
                /(beta-h*lambda2);
            v = (lambda2*(U(i-1,j) + U(i+1,j)) ...
                + U(i,j-1) + U(i,j+1))/beta;
            U(nx+1,j) = (2*lambda2*U(nx,j) + ...
                U(nx+1,j-1) + U(nx+1,j+1))/beta2;
            incr = max(incr,abs(U(i,j)-v));
            U(i,j) = v;
        end
    end
    iter = iter+1;
end
```

Al llamar al fichero

```
>> [U,iter] = GSEDP1(-1,1,-1,1,.25,.25, ...  
    inline('1+0*x'), inline('0*x'), .00001,200),
```

obtenemos, tras 170 iteraciones, la estimación de u tabulada en los nodos (x_i, y_j) .

Mostramos algunos de estos resultados en la Tabla 6.

i	$u(x_i, -1)$	$u(x_i, -0.5)$	$u(x_i, 0)$	$u(x_i, 0.5)$	$u(x_i, 1)$
0	1	1.121929	0.921286	0.517137	0
1	1	1.019062	0.831645	0.466834	0
2	1	0.969829	0.787006	0.442498	0
3	1	0.961091	0.783852	0.443206	0
4	1	0.989013	0.822892	0.470209	0
5	1	1.058027	0.910157	0.527349	0
6	1	1.181760	1.057971	0.621463	0
7	1	1.386767	1.285863	0.762708	0
8	1	1.719086	1.620880	0.964777	0

Tabla 6: Aproximaciones por el método de Gauss-Seidel

Podemos observar la matriz completa representada en la Figura 1.

8.4 Ejercicios propuestos

Ejercicio 1. Consideremos la ecuación de Helmholtz

$$u_{xx} + u_{yy} = -4u, \quad 0 < x < 1, \quad 0 < y < 1,$$

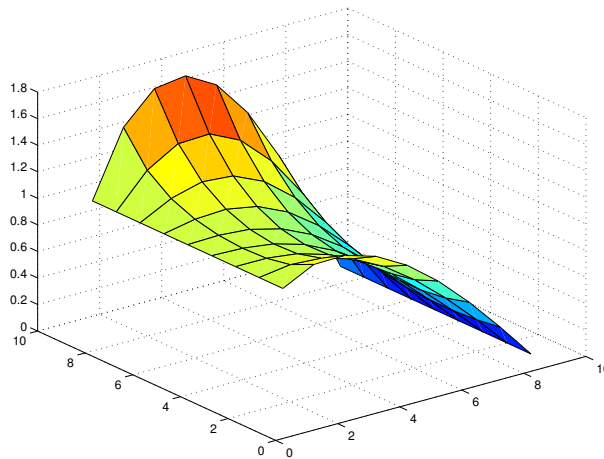


Figura 1: Aproximación de U por Gauss-Seidel

con las condiciones de contorno

$$\begin{aligned} u(x, 0) &= \cos(2x), & u(x, 1) &= \cos(2x) + \sin(2), & x &\in [0, 1], \\ u(0, y) &= \sin(2y) + 1, & u(1, y) &= \sin(2y) + \cos 2, & y &\in [0, 1]. \end{aligned}$$

- Discretiza el problema tomando $h = 0.2$ y $k = 0.1$ como pasos en x e y , respectivamente.
- Resuelve el sistema mediante el método de relajación, para distintos valores del parámetro w .
- Comprueba que la solución exacta es $u(x, y) = \sin(2y) + \cos(2x)$. Calcula el error máximo cometido en las aproximaciones del apartado b).

Ejercicio 2. Consideremos la ecuación de Poisson en coordenadas polares

$$u_{rr} + \frac{1}{r^2}u_{\theta\theta} + \frac{1}{r}u_r = -r^2 \sin(2\theta), \quad 1 < r < 2, \quad 0 < \theta < \pi,$$

con las condiciones de contorno $u(1, \theta) = \sin(\theta)$, $u(2, \theta) = 0$, $\theta \in [0, \pi]$; $u(r, 0) = 0$, $u(r, \pi) = r - 1$, $r \in [1, 2]$.

- Transforma el problema en un esquema en diferencias de orden $O(k^2 + h^2)$.

- b) Resuelve el sistema anterior mediante el método de sobrerrelajación, tomando $h = 0.1$, $k = \pi/6$, una tolerancia de 10^{-8} y los valores del factor de relajación $w = 0.2, 0.4, \dots, 1.6, 1.8$. Representa en una gráfica la relación existente entre el valor de w y el número de iteraciones necesarias en cada caso.

Ejercicio 3. Consideremos la ecuación de Poisson

$$u_{xx} + u_{yy} = -(\cos(x+y) + \cos(x-y)), \quad 0 < x < \pi, \quad 0 < y < \pi/2,$$

con las condiciones de contorno

$$u(0, y) = \cos(y), \quad u(\pi, y) = -\cos(y), \quad 0 \leq y \leq \pi/2,$$

$$u(x, 0) = \cos(x), \quad u(x, \pi/2) = 0, \quad 0 \leq x \leq \pi.$$

- a) Utilizando diferencias finitas de orden 2, transforma el problema en un sistema lineal. Toma como pasos para x e y , $h = \pi/10$ y $k = \pi/20$ respectivamente.
- b) Determina la solución del sistema anterior iterando por el método de Jacobi hasta que la variación sea menor que 10^{-10} , partiendo de la estimación inicial nula en todos los puntos interiores.
- c) Comprueba que el factor de relajación w tiene un valor óptimo en

$$w = \frac{4}{2 + \sqrt{4 - (\cos(\pi/n) + \cos(\pi/m))^2}},$$

siendo n y m el número de subintervalos considerados en el eje x e y respectivamente. Aplica el método de sobrerrelajación con factor óptimo, para resolver el sistema del apartado a). Compara la iteraciones necesarias en ambos casos.

- d) Sabiendo que la solución exacta del problema es $u(x, y) = \cos(x) \cos(y)$, determina el error máximo de cada método utilizado.

Ejercicio 4. Considera el sistema $Ax = c$ de tamaño $n \times n$:

$$\begin{pmatrix} a & b & 0 & \cdots & 0 & 0 \\ b & a & b & \cdots & 0 & 0 \\ 0 & b & a & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a & b \\ 0 & 0 & 0 & \cdots & b & a \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}.$$

- (a) Comprueba que las iteraciones de Jacobi para este sistema convergen si $|a| - 2|b| \geq 0$ y divergen en caso contrario.
- (b) Analiza la relación entre el número de iteraciones necesarias para satisfacer determinada tolerancia y el valor de b , fijados los restantes parámetros.
- (c) Considera el sistema $Ax = c$ con $a = 3$ y $b = 1$, para valores de $n = 100, 200, \dots, 1000$. Compara los tiempos de ejecución al resolver el sistema mediante el comando `\` y utilizando el método de Jacobi, considerando las matrices como llenas y como dispersas.

Ejercicio 5. Consideremos la ecuación de Laplace

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (2)$$

donde $u(x, y)$ es la función incógnita (u es la temperatura o el potencial en un punto (x, y) del plano), en el rectángulo $[0, 1] \times [0, 1]$ con condiciones de frontera

$$u(x, 0) = u(x, 1) = 0 \quad u(0, y) = u(1, y) = y(1 - y).$$

- a) Halla el sistema lineal resultante al discretizar la ecuación de Laplace mediante diferencias simétricas, con paso $h = 0.1$ en ambas variables.
- b) Resuelve de forma directa este sistema y representa gráficamente la solución.

- c) Estudia la convergencia de los métodos de Jacobi y Gauss-Seidel para dicho sistema.
- d) Halla un factor ω que acelere la convergencia por el método de sobrerelajación.

Ejercicio 6. La distribución de potencial a lo largo de una línea de transmisión formada por dos tubos de longitud L , alimentada por una fuente de voltaje, siendo r la resistividad de los tubos transmisores y g la conductancia del terreno, verifica el siguiente problema de frontera:

$$v''(x) = rgv(x), \quad x \in [0, L],$$

$$v(L) = v_L, \quad v'(0) = 0.$$

Vamos a resolver el problema para $r = 2$, $g = 0.5$, $L = 2$ y $v_L = 1$.

- a) Utiliza la aproximación de la segunda derivada y realiza una partición del intervalo $[0, L]$ para obtener una discretización del problema reduciéndolo a un sistema lineal de tamaño 9×9 , a partir del cual obtendremos una aproximación a la solución del problema de frontera dado. Nótese que la condición de contorno $v'(0) = 0$ implica $v(0) = v(x_1)$.
- b) Resuelve el problema por un método directo. Compara la solución con la exacta $v(x) = 0.133e^x + 0.133e^{-x}$.
- c) Estudia la convergencia de los métodos de Jacobi y Gauss-Seidel.
- d) Halla un factor ω que acelere la convergencia por el método de sobrerelajación.

Ejercicio 7. Consideremos la matriz de tamaño $n \times n$, $A = (a_{ij})$, donde

$$a_{ij} = \begin{cases} i(n-j+1) & \text{si } i \leq j \\ a_{ji} & \text{si } i > j \end{cases}$$

y los términos independientes $b = (1, 2, \dots, n)^T$. Resuelve el sistema $Ax = b$, con $n = 50$, mediante

- a) El operador `\` de Matlab.
- b) El método de Jacobi.
- c) El método de Gauss-Seidel.
- d) El método de sobrerelajación.
- e) Compara la solución obtenida con los diferentes métodos.

Ejercicio 8. Una persona espera impacientemente a otra que llega con retraso, de manera que camina nerviosa a lo largo de una calle, se mueve hacia la izquierda con frecuencia triple que a la derecha. Supongamos posiciones numeradas en la calle de 0 a 20, de manera que partiendo de $x_1 = 1$ y $x_{20} = 0$ las posiciones a lo largo del tiempo se obtendrán resolviendo el siguiente sistema tridiagonal:

$$x_k = \frac{3}{4}x_{k-1} + \frac{1}{4}x_{k+1}$$

- a) Utiliza el método de Crout para resolver el sistema dado.
- a) Aplica el método iterativo de Gauss-Seidel para obtener la solución.
- a) Experimenta con distintos valores de w en el método de sobrerelajación para acelerar la convergencia.
- a) El problema descrito es un problema de frontera para una ecuación en diferencias, cuya solución exacta es:

$$x_k = 1 - \frac{3^k - 1}{3^{20} - 1}$$

Calcula estos valores para $k = 0, 1, 2, \dots, 20$ y compara los resultados con los obtenidos en los apartados anteriores.

Ejercicio 9. Mediante un estudio estadístico se ha observado que durante los últimos años el número de visitas a una determinada web es tal que sumando a la cantidad que accedió un año la tercera parte de la que lo hizo el año anterior resulta la que entró al año siguiente.

- a) Supongamos que inicialmente han visitado la web 10.000 usuarios, plantea y resuelve un sistema de ecuaciones lineales para hallar los usuarios de los tres años siguientes sabiendo que sólo en el último año el número de visitas se ha multiplicado por 5. (El sistema será de 3×3 puesto que x_1 y x_5 ya se conocen).
- a) Vamos a generalizar la situación anterior para calcular el número de usuarios de n años consecutivos, suponiendo también que partimos de una situación de 10.000 inicialmente y que en el último año el número se multiplica por n . Expresa matricialmente el sistema de tamaño $(n - 2) \times (n - 2)$.
- a) Edita una función .m que construya la matriz del sistema, los términos independientes y resuelva el sistema, mediante el método de Gauss-Seidel, para cualquier valor de n que demos como parámetro de entrada.
- a) Prueba la función .m anterior para el caso del apartado a) y comprueba que obtienes el mismo resultado.
- a) Obtén ahora el vector que da el número de usuarios para $n = 20$; Utiliza la instrucción reshape para mostrar el vector solución como una matriz de $\frac{n}{5} \times 5$.
- a) Modifica la función .m anterior de modo que se considere la matriz del sistema como dispersa para la resolución del sistema. Con ayuda de los comandos tic y toc comprueba si ahorramos tiempo al trabajar con matrices dispersas, resolviendo el sistema para valores de n grandes.

Ejercicio 10. Consideremos la ecuación de Poisson con condiciones de contorno

mixtas:

$$\begin{aligned} -\frac{1}{2}u_{xx} + 4x^2u_{yy} &= \sin(y) \sin(x^2), \quad 0 < x < \pi, \quad 0 < y < \pi, \\ u(x, 0) &= 0, & u(x, \pi) &= 0, \\ u_x(0, y) &= 0, & u_x(\pi, y) &= 0. \end{aligned} \tag{3}$$

- Transforma este problema en un sistema lineal, aplicando diferencias finitas de segundo orden con paso $h = k = \frac{\pi}{10}$ para x e y , respectivamente.
- Aplica el método de Gauss-Seidel con tolerancia $tol = 10^{-5}$ para resolver el sistema anterior. ¿Cuántas iteraciones son necesarias?
- Comprueba que la solución exacta del problema es $u(x, y) = \sin(y) \cos(x^2)$ y determinar el error exacto cometido en cada punto del mallado.