

$$1. \text{ Seu } A = \begin{pmatrix} 0 & -20 & 14 \\ -3 & 27 & 4 \\ -4 & 11 & 2 \end{pmatrix}$$

a) Aplicar las transformaciones de Householder para calcular $A = QR$

$$\beta_1 = -\| (0, -20, 14) \|_2 = -5$$

$$\omega_1 = \frac{\sqrt{2}}{\| A_1 - \beta_1 e_1 \|} = \frac{\sqrt{2}}{\| (5, -3, -4) \|_1} = \frac{\sqrt{2}}{\sqrt{50}} = \frac{1}{5}$$

$$v_1 = \frac{1}{5} (5, -3, -4) = (1, -\frac{3}{5}, -\frac{4}{5})$$

$$U_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & -3/5 & -4/5 \\ -3/5 & 9/25 & 12/25 \\ -4/5 & 12/25 & 16/25 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & -3/5 & -4/5 \\ -3/5 & 9/25 & 12/25 \\ -4/5 & 12/25 & 16/25 \end{pmatrix} = \begin{pmatrix} 0 & 3/5 & 4/5 \\ 3/5 & 16/25 & -12/25 \\ 4/5 & -12/25 & 9/25 \end{pmatrix}$$

$$\text{as } U_1 A = \begin{pmatrix} 0 & 3/5 & 4/5 \\ 3/5 & 16/25 & -12/25 \\ 4/5 & -12/25 & 9/25 \end{pmatrix} \begin{pmatrix} 0 & -20 & 14 \\ -3 & 27 & 4 \\ -4 & 11 & 2 \end{pmatrix} = \begin{pmatrix} -5 & 25 & 4 \\ 0 & 0 & 10 \\ 0 & -25 & 10 \end{pmatrix}$$

$$\text{ahora } \beta_2 = -\| (0, -25) \|_1 = -25$$

$$\omega_2 = \frac{\sqrt{2}}{\| (25, -25) \|_1} = \frac{\sqrt{2}}{\sqrt{1250}} = \frac{1}{\sqrt{625}} = \frac{1}{25}$$

$$v_2 = \frac{1}{25} (25, -25) = (1, -1)$$

$$U_2^+ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} (1, -1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$U_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$Y \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} -5 & 25 & 4 \\ 0 & 0 & 10 \\ 0 & -25 & 10 \end{pmatrix} = \begin{pmatrix} -5 & 25 & 4 \\ 0 & -25 & 10 \\ 0 & 0 & 10 \end{pmatrix} = R$$

assi $U_2, U_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3/5 & 4/5 \\ 2/5 & 16/25 & -12/25 \\ 4/5 & -12/25 & 9/25 \end{pmatrix} =$

$$\begin{pmatrix} 0 & 3/5 & 4/5 \\ 4/5 & -12/25 & 9/25 \\ 3/5 & 16/25 & -12/25 \end{pmatrix} = Q$$

6) Aplique Gram-Schmidt para calcular $A = QR$

$$v_1 = (0, -3, -4)$$

$$v_2 = (-20, 27, 11) - \frac{(-20, 27, 11) \cdot (0, -3, -4)}{(0, -3, -4) \cdot (0, -3, -4)} (0, -3, -4)$$

$$= (-20, 27, 11) - \frac{(-125)}{25} (0, -3, -4)$$

$$= (-20, 27, 11) + (0, -15, -20) = (-20, 12, -4)$$

$$v_3 = (14, 4, 2) - \frac{(14, 4, 2) \cdot (0, -3, -4)}{(0, -3, -4) \cdot (0, -3, -4)} (0, -3, -4)$$

$$- \frac{(14, 4, 2) \cdot (-20, 12, -4)}{(-20, 12, -4) \cdot (-20, 12, -4)} (-20, 12, -4)$$

$$= (14, 4, 2) - \frac{-120}{25} (0, -3, -4) - \frac{(-250)}{625} (-20, 12, -4)$$

$$= \left(6, \frac{32}{5}, -\frac{24}{5} \right)$$

assi $Q = \begin{pmatrix} 0 & -4/5 & 3/5 \\ -3/5 & 12/25 & 16/25 \\ -4/5 & -9/25 & -12/25 \end{pmatrix}$

$$\text{así } Q^T A = R = \begin{pmatrix} 5 & -25 & -4 \\ 0 & 25 & -10 \\ 0 & 0 & 10 \end{pmatrix}$$

c) los cálculos en mat lab coinciden mayormente.

2.

```
Editor - C:\Users\yulis\Documents\Codigo taller4\Minimos_cuadrados.m
1 clc; clear; close all;
2
3 data = load('Datos.txt'); % Cargar el archivo de texto (2 columnas: x y)
4 x = data(:,1); % Extraer valores de x
5 y = data(:,2); % Extraer valores de y
6
7 %% Matriz A
8 n = length(x);
9 m = 5;
10 A = zeros(n, m+1);
11
12 for i = 1:m+1
13     A(:, i) = x.^^(i-1);
14 end
15
16 %% Ecuaciones Normales
17 AtA = A.' * A;
18 AtY = A.' * y;
19 c_normal = AtA \ AtY;
20
21 %% Factorización QR
22 [Q, R] = qr(A);
23 c_qr = R \ (Q' * y);
24
25 %% Verdaderos
26 c_certificados = ones(m+1, 1);
27
28 % Residuo
29 residual_normal = norm(A * c_normal - y, 2);
30 residual_qr = norm(A * c_qr - y, 2);

Editor - C:\Users\yulis\Documents\Codigo taller4\Minimos_cuadrados.m
31
32 %% Errores relativos
33 error_relativo_normal = abs((c_normal - c_certificados) ./ c_certificados);
34 error_relativo_qr = abs((c_qr - c_certificados) ./ c_certificados);
35
36 %% Resultados
37 disp('Coeficientes obtenidos (Ecuaciones Normales):');
38 disp(c_normal);
39 disp('Coeficientes obtenidos (QR):');
40 disp(c_qr);
41 disp('Coeficientes certificados:');
42 disp(c_certificados);
43
44 disp('Residual ||Ac - y||_2 (Ecuaciones Normales):');
45 disp(residual_normal);
46 disp('Residual ||Ac - y||_2 (QR):');
47 disp(residual_qr);
48
49 disp('Diferencia relativa (Ecuaciones Normales):');
50 disp(error_relativo_normal);
51 disp('Diferencia relativa (QR):');
52 disp(error_relativo_qr);
53
54 %% Graficas
55 x_fit = linspace(min(x), max(x), 100); % Puntos para la curva ajustada
56 y_fit_normal = polyval(flip(c_normal'), x_fit); % Evaluar polinomio con ecuaciones normales
57 y_fit_qr = polyval(flip(c_qr'), x_fit); % Evaluar polinomio con QR
58

Editor - C:\Users\yulis\Documents\Codigo taller4\Minimos_cuadrados.m
59 figure;
60 scatter(x, y, 'bo', 'filled'); hold on; % Datos originales
61 plot(x_fit, y_fit_normal, 'r-', 'LineWidth', 2); % Ajuste con ecuaciones normales
62 plot(x_fit, y_fit_qr, 'g--', 'LineWidth', 2); % Ajuste con QR
63 legend('Datos', 'Ajuste (Ecuaciones Normales)', 'Ajuste (QR)');
64 xlabel('x');
65 ylabel('y');
66 title('Ajuste Polinomial de Grado 5');
67 grid on;
```

Coeficientes obtenidos (Ecuaciones Normales):

1.0000
1.0000
1.0000
1.0000
1.0000
1.0000

Coeficientes obtenidos (QR):

1.0000
1.0000
1.0000
1.0000
1.0000
1.0000

Coeficientes certificados:

1
1
1
1
1
1

Residual ||Ac - y||_2 (Ecuaciones Normales):

9.1408e+05

```

Command Window
1

Residual ||Ac - y||_2 (Ecuaciones Normales):
9.1408e+05

Residual ||Ac - y||_2 (QR):
9.1408e+05

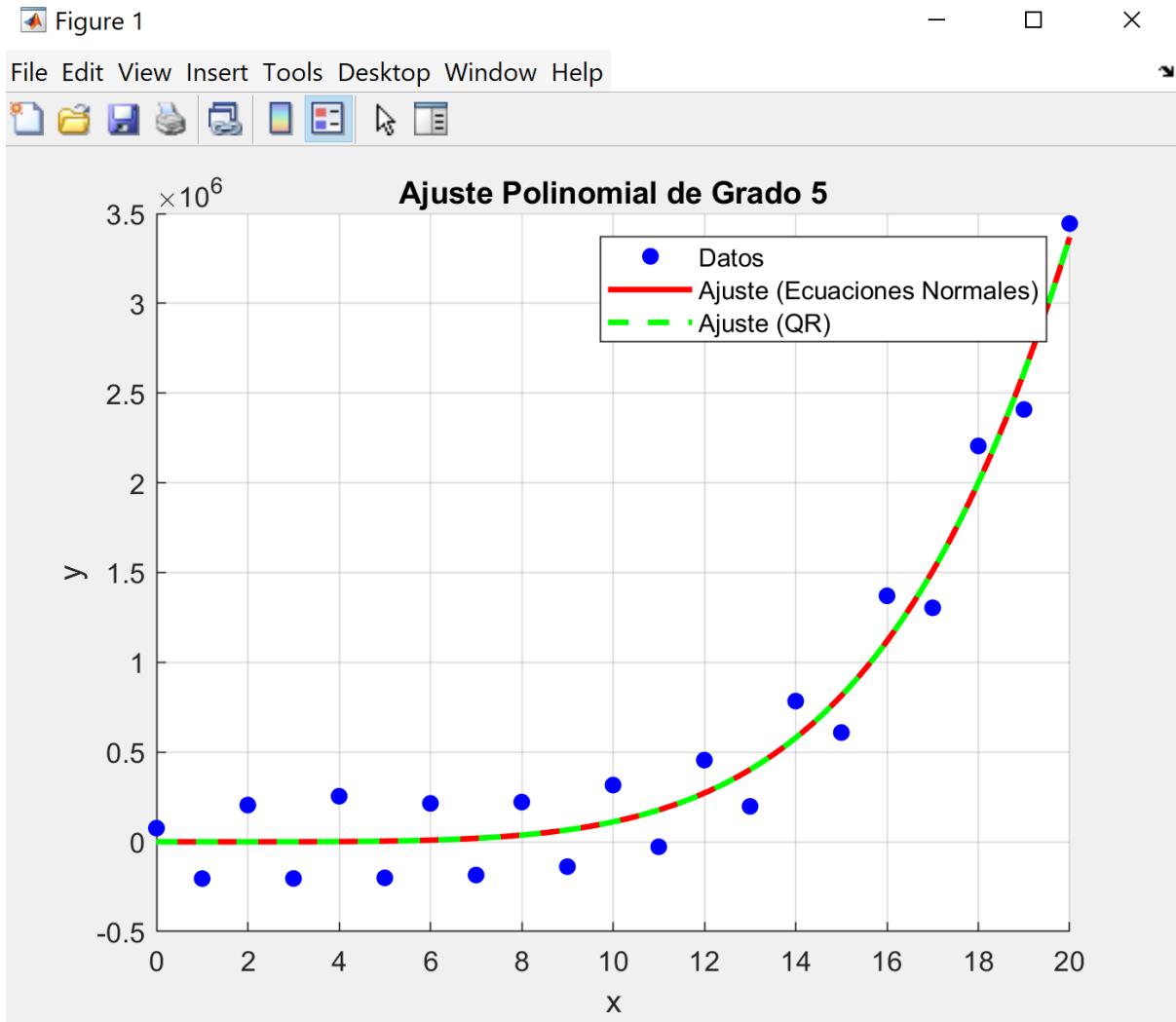
Diferencia relativa (Ecuaciones Normales):
1.0e-06 *

0.0952
0.2261
0.0880
0.0122
0.0007
0.0000

Diferencia relativa (QR):
1.0e-07 *

0.1365
0.2475
0.0864
0.0112
0.0006
0.0000

```



De lo anterior, concluimos que como la diferencia relativa para la factorización QR es menor entonces es un método más efectivo que el de las ecuaciones normales.

3. Sean

$$A = \begin{pmatrix} a & 0 & 0 \\ a\delta & a & 0 \\ 0 & a\delta & a \end{pmatrix}, \quad a < 0, \quad \delta > 0 \quad b = \begin{pmatrix} -1 \\ -1.1 \\ 0 \end{pmatrix}$$

- (a) Obtenga el número de condición de A .
 (b) Estudiar el condicionamiento del sistema $Ax = b$ en función de los valores de δ . Interprete su resultado.
 (c) Si $a = -1$, $\delta = 0.1$ y se considera $x^* = (1, 9/10, 1)^t$ como solución aproximada del sistema $Ax = b$ (sin obtener la solución exacta) un intervalo en el que esté comprendido el error relativo. ¿Es coherente con la respuesta dada en el apartado anterior?
 (d) Si $a = -1$ y $\delta = 0.1$, ¿es convergente el método de Jacobi aplicado a la resolución del sistema $Ax = b$? Realice tres iteraciones a partir de $x^0 = (0, 0, 0)^t$

a) $\kappa(A) = \|A\| \cdot \|A^{-1}\|$

valores propios son $\lambda_1 = \lambda_2 = \lambda_3 = a$

$$\|A\| = (\rho(AA^T))^{\frac{1}{2}}$$

$$AA^T = \begin{pmatrix} a & 0 & 0 \\ a\delta & a & 0 \\ 0 & a\delta & a \end{pmatrix} \begin{pmatrix} a & a\delta & 0 \\ 0 & a & a\delta \\ 0 & 0 & a \end{pmatrix} = \begin{pmatrix} a^2 & a^2\delta & 0 \\ a^2\delta & a^2\delta^2+a^2 & a^2\delta \\ 0 & a^2\delta & a^2\delta^2+a^2 \end{pmatrix}$$

$$\det(AA^T - \lambda I) = 0 \implies \lambda_1 = a^2, \quad \lambda_2 = a^2(1+\delta^2), \quad \lambda_3 = a^2(1+\delta^2+\delta^4)$$

$$\|A\|_2 = \sqrt{\lambda_3} = |a| \sqrt{1 + \delta^2 + \delta^4}$$

$$A^{-1} = \begin{pmatrix} \frac{1}{a} & 0 & 0 \\ -\frac{\delta}{a} & \frac{1}{a} & 0 \\ \frac{\delta^2}{a} & -\frac{\delta}{a} & \frac{1}{a} \end{pmatrix}$$

$$A^{-T} A^{-1} = \begin{pmatrix} \frac{1}{a} & -\frac{\delta}{a} & \frac{\delta}{a} \\ 0 & \frac{1}{a} & -\frac{\delta}{a} \\ 0 & 0 & \frac{1}{a} \end{pmatrix} \begin{pmatrix} \frac{1}{a} & 0 & 0 \\ -\frac{\delta}{a} & \frac{1}{a} & 0 \\ \frac{\delta^2}{a} & -\frac{\delta}{a} & \frac{1}{a} \end{pmatrix} = \begin{pmatrix} \frac{1 + \delta^2 + \delta^4}{a^2} & -\frac{\delta(1 + \delta^2)}{a^2} & \frac{\delta^2}{a^2} \\ -\frac{\delta(1 + \delta^2)}{a^2} & \frac{1 + \delta^2}{a^2} & -\frac{\delta}{a^2} \\ \frac{\delta^2}{a^2} & -\frac{\delta}{a^2} & \frac{1}{a^2} \end{pmatrix}$$

$$A^{-T} = \begin{pmatrix} \frac{1}{a} & -\frac{\delta}{a} & \frac{\delta^2}{a} \\ 0 & \frac{1}{a} & -\frac{\delta}{a} \\ 0 & 0 & \frac{1}{a} \end{pmatrix}$$

$$\det(A^{-T} A^{-1} - \lambda I) = 0 \implies \mu_1 = \frac{1}{a^2}, \quad \mu_2 = \frac{1}{a^2(1+\delta^2)}, \quad \mu_3 = \frac{1}{a^2(1+\delta^2+\delta^4)}$$

$$\|A^{-1}\| = \sqrt{\mu_1} = \frac{1}{|a|}$$

$$\Rightarrow \kappa(A) = |a| \sqrt{1 + \delta^2 + \delta^4} \cdot \frac{1}{|a|} = \sqrt{1 + \delta^2 + \delta^4}$$

b) $Ax = b$ $\kappa(A)$ depende de δ

- $\delta \rightarrow 0$: $\kappa(A) \rightarrow 1 \rightarrow A$ bien condicionado
- $\delta \rightarrow \infty$: $\kappa(A) \rightarrow \infty \rightarrow A$ mal condicionado

$$c) \quad a = -1 \quad b = 0,1 \quad K^+ = \begin{pmatrix} 1 \\ 0,9 \\ 1 \end{pmatrix} \quad A = \begin{pmatrix} -1 & 0 & 0 \\ -0,1 & -1 & 0 \\ 0 & -0,1 & -1 \end{pmatrix} \quad b = \begin{pmatrix} -1 \\ -1,1 \\ 0 \end{pmatrix}$$

$$r = b - Ax$$

$$Ax^+ = \begin{pmatrix} -1 \\ -1 \\ -1,09 \end{pmatrix} \Rightarrow r = \begin{pmatrix} 0 \\ -0,1 \\ 1,09 \end{pmatrix}$$

error relativo

$$\frac{\|x - x^+\|}{\|x\|} \leq K(A) \cdot \frac{\|r\|}{\|b\|} \quad K(A) = \sqrt{1 + (-0,1)^2 + 0,1^4} = \sqrt{1,1 + 0,01 + 0,0001} = \sqrt{1,0101} \approx 1,005$$

$$\|r\| = \sqrt{0^2 + (-0,1)^2 + 1,09^2} = \sqrt{0,01 + 1,1881} = \sqrt{1,1981}$$

$$\|b\| = \sqrt{1 + 1,21 + 0} = \sqrt{2,21}$$

$$K(A) \cdot \frac{\|r\|}{\|b\|} = \sqrt{1,0101} \cdot \frac{\sqrt{1,1981}}{\sqrt{2,21}} \approx 0,7400$$

error relativo está en el intervalo $[0; 0,7400]$ que es coherente con la respuesta de b

d) método de Jacobi converge cuando $\rho(T_j) < 1$

$$T = D^{-1}(L+U) \quad D = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad L+U = \begin{pmatrix} 0 & 0 & 0 \\ -0,1 & 0 & 0 \\ 0 & -0,1 & 0 \end{pmatrix}$$

$$= D^{-1}$$

$$\rightarrow T = \begin{pmatrix} 0 & 0 & 0 \\ 0,1 & 0 & 0 \\ 0 & 0,1 & 0 \end{pmatrix} \quad \text{los valores propios son } \lambda_1 = \lambda_2 = \lambda_3 = 0 < 1$$

\Rightarrow método converge

iteraciones:

$$1. \quad x_1^1 = \frac{1}{-1} (-1 - 0 \cdot 0 - 0 \cdot 0) = 1$$

$$x_1^3 = \frac{1}{-1} (-1 - 0 \cdot 1 - 0 \cdot 0,11) = 1$$

$$x_2^1 = \frac{1}{-0,1} (-1,1 - (-0,1) \cdot 0 - 0 \cdot 0) = 1,1$$

$$x_2^3 = \frac{1}{-0,1} (-1,1 - (-0,1) \cdot 1 - 0 \cdot 0,11) = 1$$

$$x_3^1 = \frac{1}{-1} (0 - 0 \cdot 0 - (-0,1) \cdot 0) = 0$$

$$x_3^3 = \frac{1}{-1} (0 - 0 \cdot 1 - (-0,1) \cdot 1) = 0,1$$

$$2. \quad x_1^2 = \frac{1}{-1} (-1 - 0 \cdot 1,1 - 0 \cdot 0) = 1$$

$$x_2^2 = \frac{1}{-0,1} (-1,1 - (-0,1) \cdot 1 - 0 \cdot 0) = 1$$

$$x_3^2 = \frac{1}{-1} (0 - 0 \cdot 1 - (-0,1) \cdot 1,1) = 0,11$$

$$\Rightarrow x^3 = \begin{pmatrix} 1 \\ 1 \\ 0,1 \end{pmatrix}$$

4. Sea $A \in \mathbb{R}^{n \times n}$. Probar que $\lambda = 1$ es un vector propio de la matriz de iteración del método de Jacobi (o Gauss-Seidel) de A si y solo si A no es invertible.

Jacobi

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

$\exists x \neq 0$ tal que $Ax=0$

$\lambda = 1 \rightarrow \pi$ no es invertible

$\lambda = 1$ valor propio de $T_J \Rightarrow \exists v \neq 0$ tal que $T_J v = v \Rightarrow D^{-1}(L+U)v = v$

$$Lv + Uv = Dv$$

$$0 = Dv - Lv - Uv = (D - L - U)v = Av$$

$\Rightarrow \pi$ no es invertible

π no es invertible $\Rightarrow \exists$ valor propio de T_J tal que $\lambda = 1$

\downarrow
 \exists vector x tal que $\pi x = 0$

$$\pi x = (D - L - U)x = 0$$

$$Dx - Lx - Ux = 0$$

$$(L+U)x = Dx$$

$$\underbrace{D^{-1}(L+U)x}_{{T_J}} = x \quad \stackrel{\text{valor propio de } T_J}{\Rightarrow} \lambda = 1$$

$$T_{GS} = (D-L)^{-1}U$$

"

$$(D-L)^{-1}Uv = v$$

$$Uv = (D-L)v = Dv - Lv$$

$$0 = Dv - Lv - Uv = (D - L - U)v = Av \Rightarrow \pi \text{ no es invertible}$$

" \Leftarrow " $\exists x \neq 0$ tal que $Ax = 0$

$$Ax = (D - L - U)x = Dx - Lx - Ux = 0$$

$$Ux = Dx - Lx = (D - L)x$$

$$(D - L)^{-1}Ux = x \quad \stackrel{\text{valor propio de } T_{GS}}{\Rightarrow} \lambda = 1$$

Gauss Seidel

5.

<pre> Editor - C:\Users\yulis\Documents\Codigo taller4\Convergencia.m 1 clc; clear; close all; 2 3 %% Matriz A y el vector b 4 A = [3 -1 -1 0 0; 5 -1 4 0 -2 0; 6 -1 0 3 -1 0; 7 0 -2 -1 5 -1; 8 0 0 -1 2]; 9 10 b = [2; 11 -26; 12 3; 13 47; 14 -10]; 15 16 n = length(b); 17 18 %% Matriz de los métodos de Jacobi, Gauss-Seidel y Sobrerelajación 19 D = diag(diag(A)); % Matriz diagonal de A 20 L = tril(A, -1); % Parte triangular inferior de A 21 U = triu(A, 1); % Parte triangular superior de A 22 23 J = -D \ (L + U); % Matriz de Jacobi 24 S = -(D + L) \ U; % Matriz de Gauss-Seidel 25 26 rho_J = max(abs(eig(J))); % Radio espectral de Jacobi 27 rho_S = max(abs(eig(S))); % Radio espectral de Gauss-Seidel 28 29 fprintf('Radio espectral de Jacobi: %.4f\n', rho_J); 30 fprintf('Radio espectral de Gauss-Seidel: %.4f\n', rho_S); </pre>	
<pre> Editor - C:\Users\yulis\Documents\Codigo taller4\Convergencia.m 31 32 %% parámetro óptimo de sobrerelajación w* 33 omega_opt = 2 / (1 + sqrt(1 - rho_S^2)); 34 fprintf('Parámetro óptimo de sobrerelajación w*: %.2f\n', omega_opt); 35 36 %% Cantidad de iteraciones 37 tol = 1e-6; % Tolerancia 38 max_iter = 1000; 39 40 % Método de Gauss-Seidel 41 [x_gs, iter_gs] = gauss_seidel(A, b, tol, max_iter); 42 43 % Método de Sobrerelajación con w* 44 [x_sor, iter_sor] = sor(A, b, omega_opt, tol, max_iter); 45 46 fprintf('Iteraciones necesarias en Gauss-Seidel: %d\n', iter_gs); 47 fprintf('Iteraciones necesarias en Sobrerelajación (w*): %d\n', iter_sor); 48 49 iter_diff = iter_gs - iter_sor; 50 fprintf('Reducción de iteraciones con Sobrerelajación: %d menos que Gauss-Seidel\n', iter_diff); 51 52 %% Mejora en tolerancia 53 tol = 1e-7; 54 % Método de Gauss-Seidel 55 [x_gs, iter_gs] = gauss_seidel(A, b, tol, max_iter); 56 57 % Método de Sobrerelajación con w* 58 [x_sor, iter_sor] = sor(A, b, omega_opt, tol, max_iter); 59 60 fprintf('Iteraciones necesarias para mejorar en un decimal en Gauss-Seidel: %d\n', iter_gs); 61 fprintf('Iteraciones necesarias para mejorar en un decimal en Sobrerelajación (w*): %d\n', iter_sor); </pre>	

```

Editor - C:\Users\yulis\Documents\Codigo taller4\Convergencia.m
63
64 %% Función para el método de Gauss-Seidel
65 function [x, iter] = gauss_seidel(A, b, tol, max_iter)
66     n = length(b);
67     x = zeros(n,1); % Solución inicial
68     iter = 0;
69
70     for k = 1:max_iter
71         x_old = x;
72         for i = 1:n
73             sum1 = A(i,1:i-1) * x(1:i-1);
74             sum2 = A(i,i+1:n) * x_old(i+1:n);
75             x(i) = (b(i) - sum1 - sum2) / A(i,i);
76         end
77         iter = iter + 1;
78         if norm(x - x_old, inf) < tol
79             break;
80         end
81     end
82 end
83

Editor - C:\Users\yulis\Documents\Codigo taller4\Convergencia.m
84 %% Función para el método de Sobrerelajación (SOR)
85 function [x, iter] = sor(A, b, omega, tol, max_iter)
86     n = length(b);
87     x = zeros(n,1); % Solución inicial
88     iter = 0;
89
90     for k = 1:max_iter
91         x_old = x;
92         for i = 1:n
93             sum1 = A(i,1:i-1) * x(1:i-1);
94             sum2 = A(i,i+1:n) * x_old(i+1:n);
95             x(i) = (1 - omega) * x_old(i) + (omega * (b(i) - sum1 - sum2) / A(i,i));
96         end
97         iter = iter + 1;
98         if norm(x - x_old, inf) < tol
99             break;
100        end
101    end
102 end

```

Command Window

```

Radio espectral de Jacobi: 0.7158
Radio espectral de Gauss-Seidel: 0.5123
Parámetro óptimo de sobrerelajación  $\omega^*$ : 1.08
Iteraciones necesarias en Gauss-Seidel: 25
Iteraciones necesarias en Sobrerelajación ( $\omega^*$ ): 20
Reducción de iteraciones con Sobrerelajación: 5 menos que Gauss-Seidel
Iteraciones necesarias para mejorar en un decimal en Gauss-Seidel: 28
Iteraciones necesarias para mejorar en un decimal en Sobrerelajación ( $\omega^*$ ): 23
fx>>

```

- Tanto el método de Gauss-Seidel como el método de Jacobi, son convergentes cuando el radio espectral de sus matrices asociadas es menor a 1, mientras que el método SOR converge siempre que Gauss-Seidel lo haga, pues su valor óptimo se calcula en función de $\rho(S)$, y si este es mayor a 1 no se puede calcular el valor óptimo, pues obtenemos una raíz de un número negativo y el valor óptimo siempre es real.
- $\rho(J)=0.7158$ y $\rho(S)=0.5123$
- $\omega^*=1.08$
- Como Gauss-Seidel tiene 5 operaciones más, esta es la reducción en el costo operacional.
- Para realizar una mejora en la precisión, Gauss-Seidel necesitó 3 iteraciones más, al igual que el método SOR.

6. Deducción del Laplaciano por Diferencias Finitas

El operador laplaciano aparece en muchas aplicaciones en física, ingeniería y procesamiento de imágenes. Su discretización mediante diferencias finitas permite resolver ecuaciones diferenciales parciales numéricamente.

Definición del Laplaciano

El operador laplaciano en dos dimensiones está definido como:

$$\nabla^2 u = \partial^2 u / \partial x^2 + \partial^2 u / \partial y^2$$

Para discretizar esta ecuación, usamos la aproximación por diferencias finitas en una malla regular.

Aproximación por Diferencias Finitas

Usando la aproximación de Taylor alrededor de un punto (i, j) en una malla con paso h , tenemos:

$$u(i+1, j) = u(i, j) + h(\partial u / \partial x) + (h^2/2)(\partial^2 u / \partial x^2) + O(h^3)$$
$$u(i-1, j) = u(i, j) - h(\partial u / \partial x) + (h^2/2)(\partial^2 u / \partial x^2) + O(h^3)$$

Sumando ambas ecuaciones y despejando la segunda derivada obtenemos:

$$\partial^2 u / \partial x^2 \approx (u(i+1, j) - 2u(i, j) + u(i-1, j)) / h^2$$

De manera análoga en la dirección y:

$$\partial^2 u / \partial y^2 \approx (u(i, j+1) - 2u(i, j) + u(i, j-1)) / h^2$$

Expresión Final del Laplaciano Discreto

Sustituyendo las aproximaciones anteriores en la ecuación del laplaciano, obtenemos:

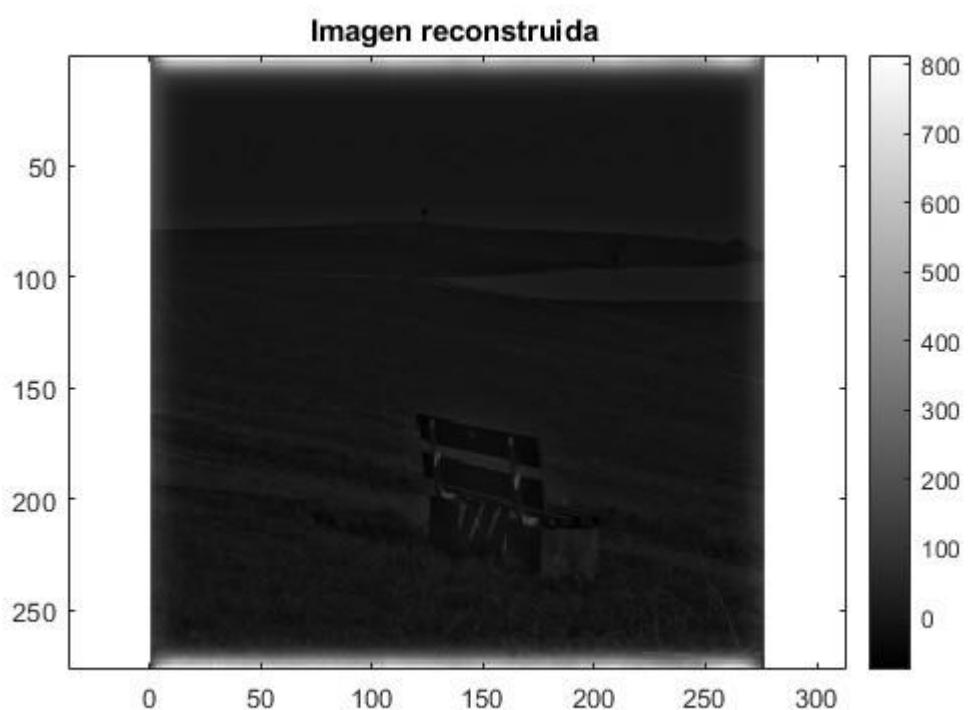
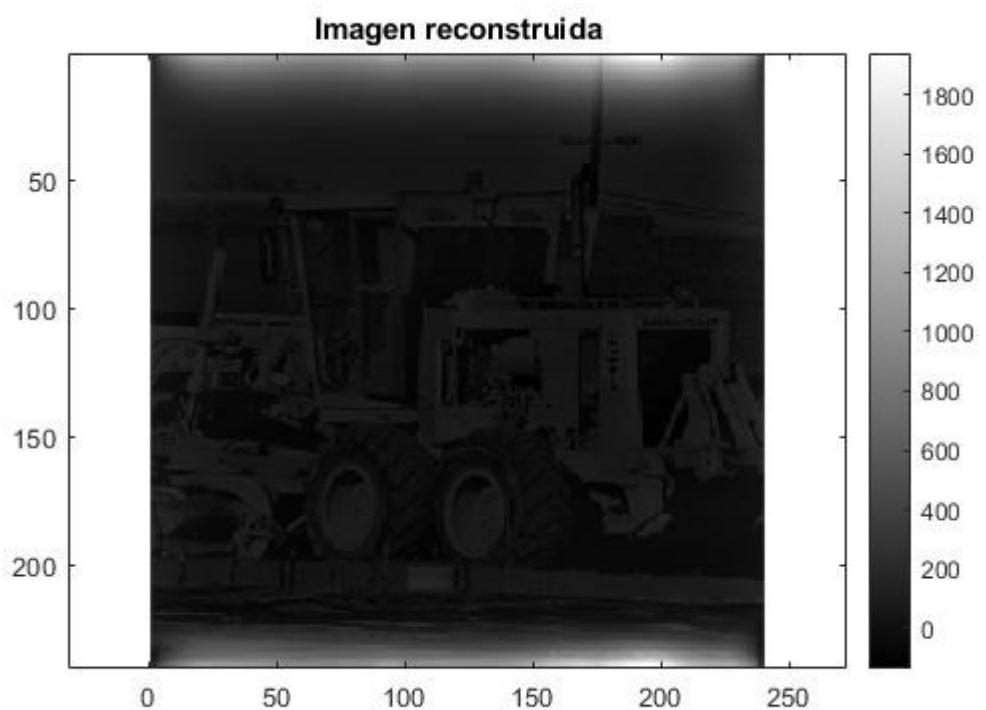
$$\nabla^2 u(i, j) \approx (u(i+1, j) + u(i-1, j) + u(i, j+1) + u(i, j-1) - 4u(i, j)) / h^2$$

Esta es la ecuación de diferencias finitas usada en métodos numéricos para resolver ecuaciones como la ecuación de Poisson.

En particular para nuestros calculos asumimos h=1; y formamos un Sistema tridiagonal siguiente, para modelar la ecuacion de poisson con $Au=b$

$$A = \begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix}$$

Y aplicando los metodos iterativos se obtuvieron las imagenes



Sergio Moreno
Marike Sandfort
Valentina Aula.