1. Consider the system $Ax = b$,

$$A = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}, \qquad b = \begin{pmatrix} 5 \\ 5 \end{pmatrix}.$$

   (a) Take two steps of the Jacobi and the Gauss–Seidel methods, starting from $x_0 = \mathbf{0}$.

   (b) Find the iteration matrix B and compute $\|B\|_\infty$ and $\rho(B)$ for each method.

   (c) Take two steps of optimal SOR iteration starting from $x_0 = \mathbf{0}$.

   (d) Find the optimal SOR iteration matrix B_{ω^*} and compute $\|B_{\omega^*}\|_\infty$, $\|B_{\omega^*}\|_1$ and $\rho(B_{\omega^*})$.

Solution:

For Jacobi method:
$$x_{k+1} = -D^{-1}(L+U)x_k + D^{-1}b$$

where

$$B_J = -D^{-1}(L+U) = \begin{pmatrix} 0 & -\frac{1}{4} \\ -\frac{1}{4} & 0 \end{pmatrix}$$

$$\|B_J\|_\infty = \max \left\{ \sum_{j=1}^{n} |b_{ij}| : 1 \le i \le n \right\} = \frac{1}{4}$$

The eigenvalues of B_J are $\lambda_1 = \frac{1}{4}$ and $\lambda_2 = -\frac{1}{4}$, so

$$\rho(B_J) = \frac{1}{4}$$

.

So,

$$x_{k+1} = \begin{pmatrix} 0 & -\frac{1}{4} \\ -\frac{1}{4} & 0 \end{pmatrix} x_k + \begin{pmatrix} \frac{5}{4} \\ \frac{5}{4} \end{pmatrix}$$

Thus, starting from $x_0 = \mathbf{0}$,

$$x_1 = \begin{pmatrix} \frac{5}{4} \\ \frac{5}{4} \end{pmatrix}, \quad x_2 = \begin{pmatrix} \frac{15}{16} \\ \frac{15}{16} \end{pmatrix}$$

For Gauss–Seidel method:

$$x_{k+1} = -(D+L)^{-1}U x_k + (D+L)^{-1}b$$

where

$$B_{GS} = -(D+L)^{-1}U = \begin{pmatrix} 0 & -\frac{1}{4} \\ 0 & \frac{1}{16} \end{pmatrix}$$

$$\|B_{GS}\|_\infty = \max \left\{ \sum_{j=1}^{n} |b_{ij}| : 1 \le i \le n \right\} = \frac{1}{4}$$

The eigenvalues of B_{GS} are $\lambda_1 = 0$ and $\lambda_2 = \frac{1}{16}$, so

$$\rho(B_{GS}) = \frac{1}{16}$$

So,

$$x_{k+1} = \begin{pmatrix} 0 & -\frac{1}{4} \\ 0 & \frac{1}{16} \end{pmatrix} x_k + \begin{pmatrix} \frac{5}{4} \\ \frac{15}{16} \end{pmatrix}$$

Thus, starting from $x_0 = \mathbf{0}$,

$$x_1 = \begin{pmatrix} \frac{5}{4} \\ \frac{15}{16} \end{pmatrix}, \quad x_2 = \begin{pmatrix} \frac{65}{64} \\ \frac{255}{256} \end{pmatrix}$$

For SOR Method:
$$(\omega L + D)x_{k+1} = [(1-\omega)D - \omega U]x_k + \omega b$$

where

$$B_{SOR} = (\omega L + D)^{-1}[(1-\omega)D - \omega U], \quad \omega = \omega^* = \frac{2}{1 + \sqrt{1 - [\rho(B_J)]^2}} = \frac{2}{1 + \sqrt{1 - \frac{1}{16}}} = \frac{8}{4 + \sqrt{15}}$$

.

So

$$B_{\omega^*} = \begin{pmatrix} -31 + 8\sqrt{15} & -8 + 2\sqrt{15} \\ 488 - 126\sqrt{15} & 93 - 24\sqrt{15} \end{pmatrix} \approx \begin{pmatrix} -0.01613 & -0.25403 \\ 0.00410 & 0.04840 \end{pmatrix}.$$

$$\|B_{\omega^*}\|_\infty = \max_i \sum_j |(B_{\omega^*})_{ij}| = 39 - 10\sqrt{15} \approx 0.2701665.$$

$$\|B_{\omega^*}\|_1 = \max_j \sum_i |(B_{\omega^*})_{ij}| = 101 - 26\sqrt{15} \approx 0.3024330.$$

The eigenvalues of B_{ω^*} are $\{\, 0,\ 31 - 8\sqrt{15} \,\}$, hence $\rho(B_{\omega^*}) = 31 - 8\sqrt{15} \approx 0.01613323$.

$$x_1 = \begin{pmatrix} 40 - 10\sqrt{15} \\ -580 + 150\sqrt{15} \end{pmatrix} \approx \begin{pmatrix} 1.2701665 \\ 0.9475019 \end{pmatrix}, \qquad x_2 = \begin{pmatrix} 6740 - 1740\sqrt{15} \\ -70100 + 18100\sqrt{15} \end{pmatrix} \approx \begin{pmatrix} 1.0089776 \\ 0.9985664 \end{pmatrix}.$$

2. Consider the strictly diagonally dominant matrices

$$A_1 = \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & -\frac{3}{4} \\ -\frac{1}{12} & 1 \end{pmatrix}.$$

Let B_1 and B_2 be the Jacobi iteration matrices of A_1 and A_2, respectively. Show that $\rho(B_1) > \rho(B_2)$. This example is intended to demonstrate that greater diagonal dominance does not imply more rapid Jacobi convergence.

$$A = D + L + U, \qquad B_J = -D^{-1}(L + U).$$

For $A_1 = \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{pmatrix}$ we have $D = I$, $L = \begin{pmatrix} 0 & 0 \\ -\frac{1}{2} & 0 \end{pmatrix}$, $U = \begin{pmatrix} 0 & -\frac{1}{2} \\ 0 & 0 \end{pmatrix}$. Hence

$$B_1 = -D^{-1}(L + U) = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}, \quad \sigma(B_1) = \{\tfrac{1}{2}, -\tfrac{1}{2}\}, \quad \rho(B_1) = \tfrac{1}{2}.$$

For $A_2 = \begin{pmatrix} 1 & -\frac{3}{4} \\ -\frac{1}{12} & 1 \end{pmatrix}$ we have $D = I$, $L = \begin{pmatrix} 0 & 0 \\ -\frac{1}{12} & 0 \end{pmatrix}$, $U = \begin{pmatrix} 0 & -\frac{3}{4} \\ 0 & 0 \end{pmatrix}$. Thus

$$B_2 = -D^{-1}(L + U) = \begin{pmatrix} 0 & \frac{3}{4} \\ \frac{1}{12} & 0 \end{pmatrix}.$$

The eigenvalues of $\begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix}$ are $\pm\sqrt{ab}$, so

$$\sigma(B_2) = \left\{\tfrac{1}{4}, -\tfrac{1}{4}\right\}, \qquad \rho(B_2) = \sqrt{\tfrac{3}{4} \cdot \tfrac{1}{12}} = \tfrac{1}{4}.$$

Therefore,

$$\rho(B_1) = \tfrac{1}{2} \ > \ \tfrac{1}{4} = \rho(B_2),$$

which shows that stronger diagonal dominance (here, A_1 has larger minimal dominance) does not necessarily yield faster Jacobi convergence.

3. With
$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix},$$

find the iteration matrices B_J and B_{GS} for the Jacobi and Gauss–Seidel methods, respectively. Find the eigenvalues of B_J and B_{GS}, and verify the relation $\rho(B_{GS}) = \rho^2(B_J)$, which holds when A is tridiagonal.

$$A = D + L + U, \qquad B_J = -D^{-1}(L+U), \qquad B_{GS} = -(D+L)^{-1}U,$$

where for $A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$ we have $D = 2I$, $L = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$, $U = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$.

Jacobi.
$$B_J = -\tfrac{1}{2}(L+U) = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{pmatrix}.$$

The eigenvalues of B_J are

$$\lambda_k = \cos\left(\frac{k\pi}{4}\right), \quad k = 1,2,3 \ \Rightarrow \ \sigma(B_J) = \left\{ \tfrac{\sqrt{2}}{2}, 0, -\tfrac{\sqrt{2}}{2} \right\}, \quad \rho(B_J) = \tfrac{\sqrt{2}}{2}.$$

Gauss–Seidel.

$$(D+L)^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}, \qquad B_{GS} = -(D+L)^{-1}U = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \\ 0 & \frac{1}{8} & \frac{1}{4} \end{pmatrix}.$$

Writing $B_{GS} = \left(\begin{smallmatrix} 0 & * \\ 0 & C \end{smallmatrix}\right)$ with $C = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} \\ \frac{1}{8} & \frac{1}{4} \end{pmatrix}$, the eigenvalues are $\{0\} \cup \sigma(C)$. Since $\mathrm{tr}(C) = \tfrac{1}{2}$ and $\det(C) = 0$,

$$\sigma(B_{GS}) = \{0, 0, \tfrac{1}{2}\}, \qquad \rho(B_{GS}) = \tfrac{1}{2}.$$

Verification.
$$\rho(B_{GS}) = \tfrac{1}{2} = \left(\tfrac{\sqrt{2}}{2}\right)^2 = \rho(B_J)^2,$$

which confirms the tridiagonal relation $\rho(B_{GS}) = \rho(B_J)^2$.

4. Let $u(x,y)$ be the steady state temperature in a square plate which is heated on one side and cooled on the remaining three sides. Then $u(x,y)$ satisfies

$$\Delta u = 0 \quad \text{for } (x,y) \in (0,1) \times (0,1),$$

with Dirichlet boundary conditions

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

Compute the solution of this boundary value problem using the 5-point Laplacian (second-order scheme) with mesh size $h = 2^{-p}$, $p = 2, 3, 4, 5$. Solve the arising linear system by Gaussian elimination. Use a contour and/or a surface plotting routine to plot the numerical solutions.

From $\Delta u = 0$, we have 5-point scheme:

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = 0$$

$$u(x,0) = 1 \implies 4u_{i,1} - u_{i-1,1} - u_{i+1,1} - u_{i,2} = u_{i,0} = 1$$

2-D to 1-D:

$$l = i + (j-1)m, \quad i = 1, \ldots, m, \ j = 1, \ldots, m$$

$$A \in \mathbb{R}^{N \times N}, \qquad A = \begin{pmatrix} T & -I_m & & & \\ -I_m & T & -I_m & & \\ & \ddots & \ddots & \ddots & \\ & & -I_m & T & -I_m \\ & & & -I_m & T \end{pmatrix}, \quad T = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{m \times m},$$

where I_m is the $m \times m$ identity and blank entries are zeros.

The right-hand side comes only from the heated bottom boundary $u(x, 0) = 1$:

$$b = \begin{pmatrix} \mathbf{1}_m \\ \mathbf{0}_m \\ \vdots \\ \mathbf{0}_m \end{pmatrix} \in \mathbb{R}^N,$$

i.e. the first m entries are 1 and the remaining $N - m$ entries are 0.

Here, $m = \frac{1}{h} - 1 = 2^p - 1$ is the number of interior grid points in each direction.

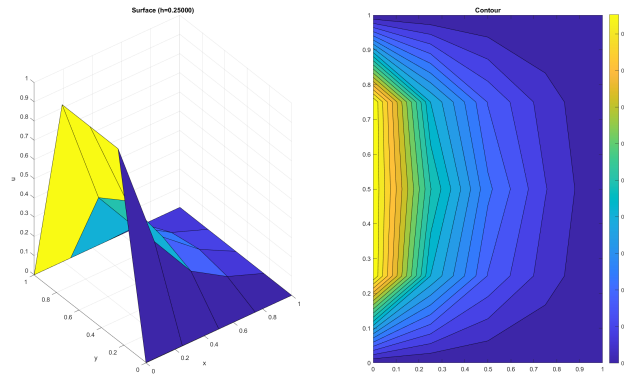**Numerical Results.**   Here are the numerical solutions for $p = 2, 3, 4, 5$:
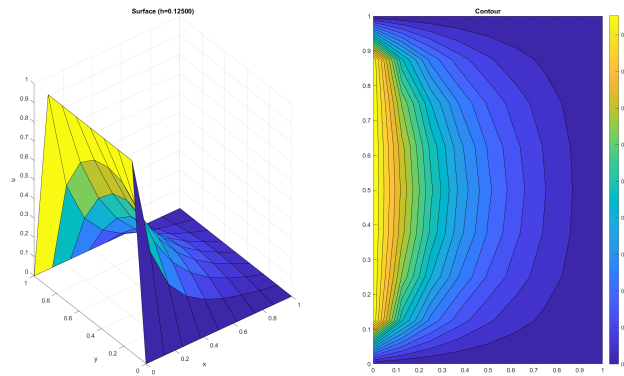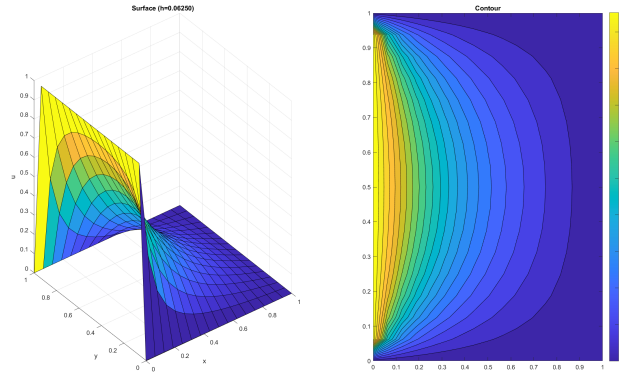

Figure 1: Numerical solution for $p = 2$ ($h = 0.25$).
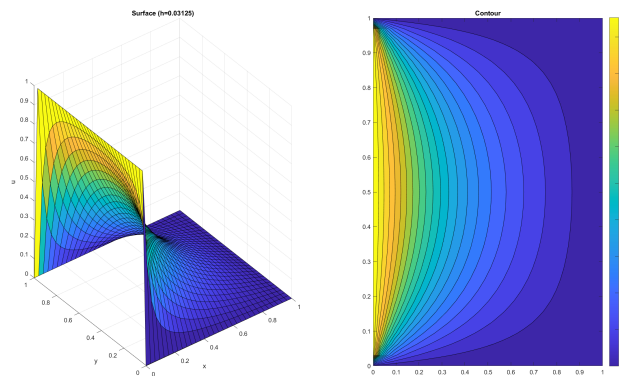

Figure 2: Numerical solution for $p = 3$ ($h = 0.125$).

Figure 3: Numerical solution for $p = 4$ ($h = 0.0625$).

Figure 4: Numerical solution for $p = 5$ ($h = 0.03125$).

5. Solve the steady state heat conduction problem from Question 4 using the Jacobi and Gauss–Seidel iterations. Take the same $h = 2^{-p}$, $p = 2, 3, 4, 5$ and start from $x^{(0)} = \mathbf{0}$. Terminate the iteration when $\|r^{(k)}\|_2 \leq 10^{-2}$, where $r^{(k)}$ is the residual at step k. Display the results for each method in a table:

- column 1: h (mesh size)
- column 2: k (total number of iterations needed to achieve error tolerance)
- column 3: $\|r^{(k)}\|_2$ (norm of residual at step k)
- column 4: $\|r^{(k)}\|_2 / \|r^{(k-1)}\|_2$ (ratio of successive residual norms at step k)

Summarize and explain the results. How does the rate of convergence behave as h reduces?

Table 1: Jacobi results

h	k (iters)	$\|r_k\|_2$	ratio
0.2500	13	9.568319×10^{-3}	7.071067×10^{-1}
0.1250	49	9.945191×10^{-3}	9.238773×10^{-1}
0.0625	166	9.887564×10^{-3}	9.807788×10^{-1}
0.0312	523	9.998456×10^{-3}	9.951703×10^{-1}

Table 2: Gauss–Seidel results

h	k (iters)	$\|r_k\|_2$	ratio
0.2500	7	7.768911×10^{-3}	4.999220×10^{-1}
0.1250	24	9.751416×10^{-3}	8.532369×10^{-1}
0.0625	81	9.695216×10^{-3}	9.617785×10^{-1}
0.0312	256	9.937994×10^{-3}	9.903053×10^{-1}

Summary and Explanation: As $h \to 0$, the total number of iterations k increases significantly for both methods, indicating slower convergence. More specifically, we have the theoretically approximated convergence factors:

$$\rho(B_J) \approx cos(\pi h) \approx 1 - \Theta(h^2)$$

As h decreases to 0, $\rho(B_J)$ approaches 1, meaning the convergence factor gets closer to 1, which is checked by the experiment.

We also have:

$$\rho(B_{GS}) = \rho(B_J)^2$$

So the convergence factors for both methods approach 1 as h decreases, confirmed by the experiment.

```matlab
1   % Laplace Dirichlet on (0,1)^2 with 5-point stencil
2   % BC: u(x,0)=1; u=0 on top/left/right
3   % Part 1: direct solve + plots (p=2..5)
4   % Part 2: Jacobi & Gauss-Seidel to ||r||_2 <= 1e-2 from x0=0; print tables
5
6   clear; clc;
7
8   ps = 2:5; tol = 1e-2;
9
10  %% ---------- Part 1: Direct solve + plots ----------
11  for p = ps
12      n = 2^p; h = 1/n; m = n-1; N = m*m;
13      e = ones(m,1);
14      T = spdiags([-e 4*e -e], -1:1, m, m); % 1D 2nd-diff (scaled)
15      S = spdiags([-e -e], [-1 1], m, m); % off-diagonal coupling
16      A = kron(speye(m),T) + kron(S,speye(m)); % 2D 5-point matrix
17
18      B = zeros(m,m); B(:,1) = 1;          % bottom boundary contributes
19      b = reshape(B,N,1);
20
21      u = A \ b;                           % Gaussian elimination (backslash)
22      U = zeros(n+1,n+1);
23      U(2:n,2:n) = reshape(u,m,m);
24      U(2:n,1)  = 1;                       % enforce BC for plotting
25
26      [X,Y] = meshgrid(0:h:1, 0:h:1);
27      figure('Name',sprintf('Direct p=%d, h=%.5f',p,h));
28      subplot(1,2,1); surf(X,Y,U); title(sprintf('Surface (h=%.5f)',h));
29      xlabel x; ylabel y; zlabel u;
30      subplot(1,2,2); contourf(X,Y,U,20); colorbar; title('Contour');
31  end
32
33  %% ---------- Part 2: Iterative solves (Jacobi & GS) ----------
34  fprintf('\nJACOBI RESULTS\n');
35  fprintf('%6s %8s %15s %15s\n','h','iters','||r_k||_2','ratio');
36  for p = ps
37      n = 2^p; h = 1/n; m = n-1; N = m*m;
38      e = ones(m,1);
39      T = spdiags([-e 4*e -e], -1:1, m, m);
40      S = spdiags([-e -e], [-1 1], m, m);
41      A = kron(speye(m),T) + kron(S,speye(m));
42
43      B = zeros(m,m); B(:,1) = 1; b = reshape(B,N,1);
44
45      D = spdiags(diag(A),0,N,N); L = tril(A,-1); Uu = triu(A,1); %#ok<NASGU>
46      Dinv = 1./diag(D);
47
48      x = zeros(N,1); r = b - A*x; rk_prev = norm(r,2);
49      k = 0; ratio = NaN; rk = rk_prev;
50      while rk > tol
51          x = x + Dinv .* (b - A*x);       % Jacobi: x^{k+1} = x^k + D^{-1} r^k
52          r = b - A*x; rk = norm(r,2);
53          k = k+1; ratio = rk / rk_prev; rk_prev = rk;
54      end
```

```matlab
55        fprintf('%6.4f %8d %15.6e %15.6e\n', h, k, rk, ratio);
56    end
57
58    fprintf('\-nGAUSSSEIDEL RESULTS\n');
59    fprintf('%6s %8s %15s %15s\n','h','iters','||r_k||_2','ratio');
60    for p = ps
61        n = 2^p; h = 1/n; m = n-1; N = m*m;
62        e = ones(m,1);
63        T = spdiags([-e 4*e -e], -1:1, m, m);
64        S = spdiags([-e -e], [-1 1], m, m);
65        A = kron(speye(m),T) + kron(S,speye(m));
66
67        B = zeros(m,m); B(:,1) = 1; b = reshape(B,N,1);
68
69        D = spdiags(diag(A),0,N,N); L = tril(A,-1); Uu = triu(A,1);
70        M = D + L;                          % GS left matrix
71
72        x = zeros(N,1); r = b - A*x; rk_prev = norm(r,2);
73        k = 0; ratio = NaN; rk = rk_prev;
74        while rk > tol
75            x = M \ (b - Uu*x);             % -GaussSeidel: (D+L)x^{k+1} = b - U x^k
76            r = b - A*x; rk = norm(r,2);
77            k = k+1; ratio = rk / rk_prev; rk_prev = rk;
78        end
79        fprintf('%6.4f %8d %15.6e %15.6e\n', h, k, rk, ratio);
80    end
81
82    % Summary (expected behavior):
83    % As h decreases, iteration counts grow ~ O(h^{-2}). GS converges faster than Jacobi
84    % (asymptotically, rho_GS  rho_J^2), but both slow down on finer meshes.
```