Let $A$ be the tridiagonal $n \times n$ matrix, with entries $a_{i,i} = 1$ for $i = 1 : n$, $a_{i,i+1} = -\alpha$ for $i = 1 : n - 1$, $a_{i,i+1} = \alpha - 1$ for $i = 1 : n - 1$. Let $b$ be the $n \times 1$ vector, with $b_1 = \alpha$ and $b_i = 0$ for $i = 2 : n$, $\alpha \in \mathbb{R}$. See Ex. 16 of our textbook (9th Edition), or (Ex. 24, pg 452 of the 8th Edition).

1. Write a MATLAB implementation for the Jacobi, Gauss-Seidel and SOR algorithms with input $\alpha$, $n$, $A$, $b$, and zero vector at start ($x^{(0)} = 0$) and with stopping criteria $\|x^{(k)} - x^{(k-1)}\|_\infty/\|x^{(1)} - x^{(0)}\|_\infty \leq 10^{-6}$.

2. For $\alpha = \frac{1}{2}$ and various $n = 10, 20, 40, 80, \cdots$, create a table indicating the number of iterations for convergence for the Jacobi and Gauss-Seidel methods.

3. For $\alpha = \frac{1}{2}$ and various $n = 10, 20, 40, 80, \cdots$, create a table indicating the number of iterations for convergence for the SOR method with $\omega = -0.4, 0.4, 0.8, 1.2, 1.6, \text{and} 2.4$. Also based on your experiments, find out which is the optimal $\omega$ for each $n$.

4. Repeat the previous experiments with $\alpha = 0.3$ and $\alpha = -0.3$.

5. Which conclusions can you obtain from your experiments?

6. Which is the largest $n$ could you run in your computer?

7. **Type your project.** Handwritten projects will not be accepted. Include your code.