Exercise 1: Poisson's Equation Consider the linear system $A_n\phi = \rho$, where A_n is an $n \times n$ matrix with 2's on the main diagonal, -1's directly above and below the main diagonal and 0's everywhere else. For instance, A_5 is

$$A_5 = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

This is a discretized version of Poisson's equation:

$$\frac{\partial^2 \phi}{\partial x^2} = \rho.$$

This equation appears very often in physics.

Construct the matrix A_{50} in Matlab. Try reading the documentation for `diag()`. Make the vector ρ according to the formula

$$\rho_j = 2\left(1 - \cos\left(23\pi/51\right)\right)\sin\left(23\pi j/51\right).$$

(a) Write down the matrix form of the Jacobi iteration $\phi_{k+1} = M\phi_k + \mathbf{c}$. Concatenate the matrix M and the vector $\mathbf{c}$ and save the resulting 50×51 matrix as **A1.dat**.

(b) Use Jacobi iteration to solve for ϕ given an initial guess of a column of ones. Continue to iterate the Jacobi method until every term in the vector ϕ is within 10^{-4} of the previous iteration. I.e.,

norm(phi(:,k+1) - phi(:,k), Inf) <= 1e-4.

Save the final iteration as a column vector in **A2.dat** and save the total number of iterations as **A3.dat**.

(c) Now write down the matrix form of the Gauss-Seidel iteration $\phi_{k+1} = M\phi_k + \mathbf{c}$. Note that these are not the same M and $\mathbf{c}$ as in part (a). Concatenate the matrix M and the vector $\mathbf{c}$ and save the resulting 50×51 matrix as **A4.dat**.

(d) Use Gauss-Seidel iteration to solve for ϕ given an initial guess of a column of ones. Continue to iterate the Gauss-Seidel method until every term in the vector ϕ is within 10^{-4} of the previous iteration (as in part (b)). Save the final iteration as a column vector in **A5.dat** and save the total number of iterations as **A6.dat**.

Exercise 2: Successive Over-Relaxation The successive over-relaxation method (SOR method) is another iterative method related to Jacobi and Gauss-Seidel iteration. Suppose we wish to solve the system $A\mathbf{x} = \mathbf{b}$, and the matrix A is decomposed into diagonal, upper, and lower parts:

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}, U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & \dots & 0 \end{bmatrix},$$

$$L = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{n,n-1} & 0 \end{bmatrix}, \qquad A = D + U + L.$$

The SOR method for a relaxation factor ω is defined as:

$$\mathbf{x}_{k+1} = (D + \omega L)^{-1}[\omega\mathbf{b} - (\omega U + (\omega - 1)D)\mathbf{x}].$$

Note that for $\omega = 1$, this is equivalent to the Gauss-Seidel method.

Consider the system $A_n\phi = \rho$ from Exercise 1.

(a) For $\omega = 1.5$, write down the matrix form of $\phi_{k+1} = M\phi_k + \mathbf{c}$. Concatenate the matrix M and the vector $\mathbf{c}$ and save the resulting 50×51 matrix as **A7.dat**.

(b) If we define the error of iteration k as $e_k = \phi - \phi_k$, we can show that the error evolves like $e_{k+1} = Be_k$. The error will decay if the eigenvalues of B are all less than one in absolute value. For $\omega = 1$ to $\omega = 1.99$ (in increments of 0.01), compute the eigenvalues of the matrix B. Save the absolute values of these eigenvalues as a 100×50 matrix in **A8.dat**. Find the choice of ω that yields the smallest maximal eigenvalue (in absolute value). Save this ω and the absolute value of its maximal eigenvalue as a column vector (with ω as the first entry) in **A9.dat**.

(c) For the same set of ω choices, compute 200 SOR iterations from an initial guess of a column of ones. Compute the residual $A_{50}\phi_{200} - \rho$ of the final iteration ϕ_{200} for each case. Save the 2-norm of each residual as a 100×1 column vector in **A10.dat**. Does the smallest residual correspond to the same optimal ω as you found in part (b)?

(d) Use SOR iteration with the optimal ω from part (b) to solve for ϕ given an initial guess of a column of ones. Continue to iterate the SOR method until every term in the vector ϕ is within 10^{-4} of the previous iteration (as in Exercise 1, part (b)). Save the final iteration as a column vector in **A11.dat** and save the total number of iterations as **A12.dat**. Which iterative method required the fewest iterations (think about this, but you do not need to submit an answer).