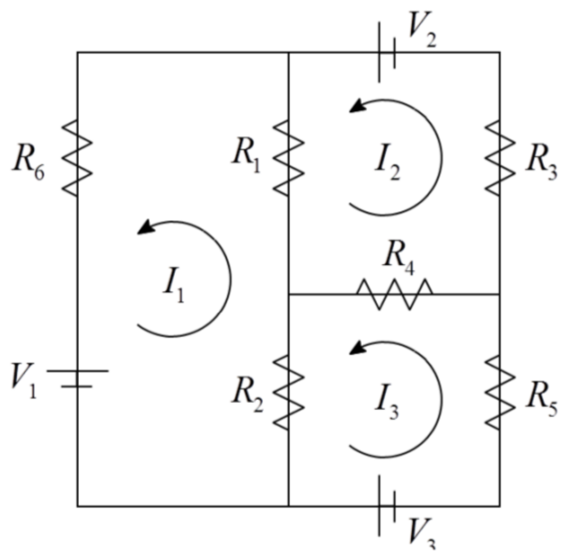
Exercise 1: LU Decomposition Consider the circuit diagram below:



Following the two rules:

(1) The voltage drop across a resistor is $V = IR$,

(2) The sum of all the voltage drops in a closed loop sum to zero,

we can construct the following systems of equations:

$$\begin{aligned}
R_6 I_1 + R_1 (I_1 - I_2) + R_2 (I_1 - I_3) &= V_1, \\
R_3 I_2 + R_4 (I_2 - I_3) + R_1 (I_2 - I_1) &= V_2, \\
R_5 I_3 + R_4 (I_3 - I_2) + R_2 (I_3 - I_1) &= V_3.
\end{aligned}$$

Let the resistances be given by $R_1 = 10\Omega$, $R_2 = 20\Omega$, $R_3 = 5\Omega$, $R_4 = 15\Omega$, $R_5 = 30\Omega$ and $R_6 = 25\Omega$.

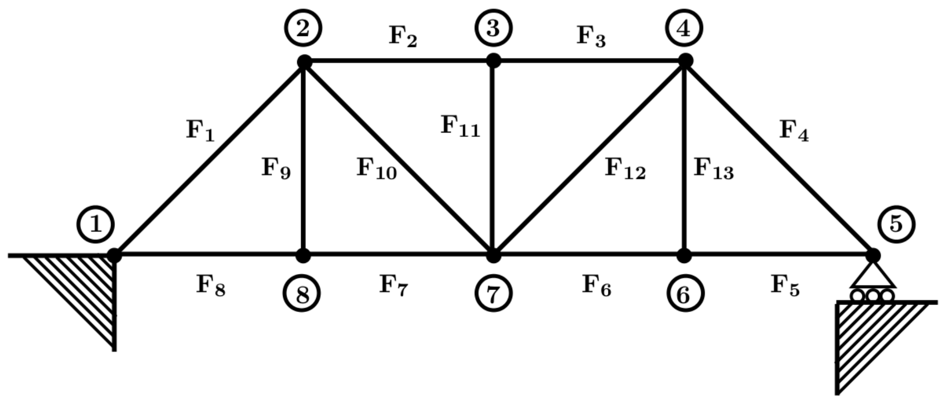
(a) Write the equations in matrix form $A\boldsymbol{x} = \boldsymbol{b}$ (you need to do this by hand). Using the lu command in matlab, find matrices L, U and P such that $PA = LU$. Concatenate A, P, L and U in one 3×12 matrix and save it as **A1.dat**.

(You don't know the voltage yet. Does that matter?)

(b) Let $V_1 = 50$, $V_2 = 0$ and let V_3 vary from 1 to 100. For each value of V_3, calculate I_1, I_2 and I_3 using P, L and U. Save all of the results in a 3×100 matrix as **A2.dat**, with the order of the columns following that of V_3.

(c) Repeat part (b), but use the inverse of A instead of L, U and P (using the command inv). This method should be slower, and the result should be slightly different. Subtract your results from parts (b) and (c), then save the absolute value of the difference as **A3.dat**. The answer should still be a 3×100 matrix.

Exercise 2: Forces on a Bridge Consider the bridge truss shown below.



Given a vector of external forces $\boldsymbol{b}$ at any of the positions 1-13, we can compute the forces $\boldsymbol{x} = [\boldsymbol{F_1}, \boldsymbol{F_2}, \dots, \boldsymbol{F_{13}}]^T$ by solving the system

$$A\boldsymbol{x} = \boldsymbol{b},$$

where A is given by

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

and $s = \sqrt{2}/2$.

We will solve for the vector of forces $\boldsymbol{x}$ assuming that there are 5 ton vehicles sitting at nodes 6, 7 and 8. This means that $\boldsymbol{b} = [0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 5, 0, 5]^T$.

(a) Solve for $\boldsymbol{x}$ using the LU-decomposition. (Use the lu command.) Save the intermediate answer $\boldsymbol{y}$ as **A4.dat** and the final answer $\boldsymbol{x}$ as **A5.dat**.

(b) Solve for $\boldsymbol{x}$ using the backslash command. Save your answer as **A6.dat**.

(c) Now suppose that we add weight to the middle truck (which corresponds to the 11th entry of $\boldsymbol{b}$) in increments of 0.01 tons until the bridge collapses. Each bridge member is rated for no more than 30 tons of compression or tension (i.e., positive or negative forces.) That is, the bridge will collapse when the absolute value of the largest force is larger than 30. Find the weight of the middle truck at the exact moment the bridge collapses. Save your answer as **A7.dat**.

    Update: You need to find the lowest weight of the middle truck such that the maximum force is greater than or equal to 30.

    Hint: You can find the absolute value of the largest entry in a vector $\boldsymbol{x}$ using the infinity norm. In matlab, this is norm(x,Inf).

Exercise 3: Poisson's Equation Consider the linear system $A_n\boldsymbol{\phi} = \boldsymbol{\rho}$, 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 diescretized 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. Make the vector $\boldsymbol{\rho}$ according to the formula

$$\boldsymbol{\rho}_j = 2\left(1 - \cos\left(23\pi/51\right)\right)\sin\left(23\pi j/51\right).$$

(a) Write down the matrix form of the Jacobi iteration $\boldsymbol{\phi}_{k+1} = M\boldsymbol{\phi}_k + \boldsymbol{c}$. Concatenate the matrix M and the vector $\boldsymbol{c}$ and save the resulting 50×51 matrix as **A8.dat**.

(b) Use Jacobi iteration to solve for $\boldsymbol{\phi}$ given an initial guess of a column of ones. Continue to iterate the Jacobi method until every term in the vector $\boldsymbol{\phi}$ 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 **A9.dat** and save the total number of iterations as **A10.dat**.

(c) Now write down the matrix form of the Gauss-Seidel iteration $\boldsymbol{\phi}_{k+1} = M\boldsymbol{\phi}_k + \boldsymbol{c}$. Note that these are not the same M and $\boldsymbol{c}$ as in part (a). Concatenate the matrix M and the vector $\boldsymbol{c}$ and save the resulting 50×51 matrix as **A11.dat**.

(d) use Gauss-Seidel iteration to solve for $\boldsymbol{\phi}$ given an initial guess of a column of ones. Continue to iterate the Gauss-Seidel method until every term in the vector $\boldsymbol{\phi}$ is within 10^{-4} of the previous iteration (as in part (b)). Save the final iteration as a column vector in **A12.dat** and save the total number of iterations as **A13.dat**.