1. Recall the Truss example in the lecture notes from Week 6. Write a Matlab program to setup and solve the linear system. Determine which beams in the truss are under tension and which are under compression.

2. Suppose you are given a Matlab data file containing a matrix $A$ and a vector $b$. In Matlab you observe:

\[
\text{>> cond}(A) \quad \text{ans} = 8.4018e+012
\]

You can solve the linear system using Matlab, with the command

\[
\text{>> x = A \backslash b}
\]

Estimate the relative error in the computed solution $x$. (Remember that when numbers are stored on the computer there is a rounding error).

3. Consider the tridiagonal matrix

\[
A = \begin{pmatrix}
2 & -1 & & \\
-1 & 2 & -1 & \\
& \ddots & \ddots & \ddots \\
& & -1 & 2 & -1 \\
& & & -1 & 2
\end{pmatrix}.
\]

This is an important matrix that will feature prominently in the material at the end of this course.

To answer this question you will find the following Matlab commands helpful: \texttt{diag}, \texttt{ones}, \texttt{lu}, \texttt{cond}, \texttt{eig}, \texttt{norm}, \texttt{max}, \texttt{min}, \texttt{tic}, \texttt{toc} and to compute the transpose of $A$ the command \texttt{A.'}.

(a) Write a Matlab function \texttt{tridiagonal(n)} so that

\[
A = \text{tridiagonal}(n)
\]

computes the $n \times n$ matrix $A$ described above.
(b) Show that the matrix $\text{tridiagonal}(10)$ is symmetric and positive definite. (Hint: a symmetric matrix is positive definite if all of its eigenvalues are greater than zero.)

(c) For a symmetric positive definite matrix $A$, the 2-norm condition number is

$$\text{cond}_2(A) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$$

where $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$ are the largest and smallest eigenvalues respectively of $A$. For $n = 10, 100, 1000$ compute the 2-norm condition number of $A$. Describe the rate of growth of the condition number as $n$ increases.

(d) Suppose we wish to solve a linear system with $A$ and we desire relative error $10^{-8}$ in our solution. If $A$ and the right hand side $b$ are stored in double precision, what is the largest $n$ for which we can get a satisfactory solution?

(e) We wish to solve the $n$ linear systems

$$Ax_k = e_k, \quad k = 1, \ldots, n,$$

when $n = 1000$. Solve these $n$ linear systems using (i) the backslash command $n$ times; and (ii) by precomputing the LU-factorisation of $A$ and then reusing $L$ and $U$. In both cases record the CPU time and explain your observations.

(f) How can the solution of these linear systems be accelerated for this matrix?