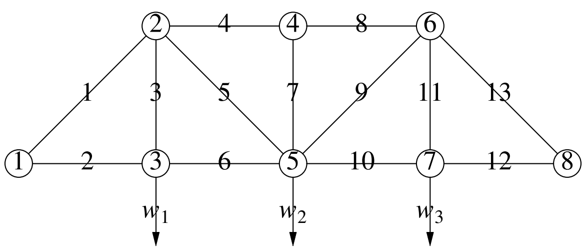
Please write your family and given names and **underline** your family name on the front page of your paper.

General note: Plotting quantity y versus quantity x, means that x is in the x-axis and y is on the y-axis, i.e. what follows "versus" is in the horizontal axis.

1. A truss is a structure consisting of vertical, horizontal and diagonal beams (members) forming one or more triangles. Trusses are used to improve the stability of buildings, bridges, etc. Members are connected at frictionless and free-to-rotate joints. The first figure below shows a truss with 13 members (numbered lines) and 8 joints (numbered circles). The second figure shows a truss with 21 members and 12 joints. Certain given loads (arrows) are applied at certain joints. It is usually assumed that the weights of the members are negligible compared to the externally applied loads. To analyze a truss, an engineer would have to calculate the forces acting along each member. (We usually assume that, when the value of the force is positive, the member is in compression, otherwise it is in tension.) Considering that the truss is at equilibrium, we can equate the sum of all forces that act on each joint horizontally to zero, do the same for the forces that act on each joint vertically, and get a system of equations, from which we can determine the forces.

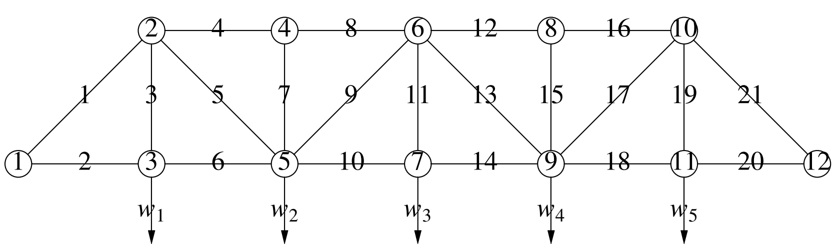
We consider trusses such as those in the figure, with n (even number) top horizontal beams, $n+2$ bottom horizontal beams, $n+1$ vertical beams and $n+2$ diagonal beams at $\frac{\pi}{4}$ angles with the horizontal and vertical beams. The number of members (and hence the number of forces, i.e. the number of unknowns) is $N = 4(n+1)+1$, while the number of joints is $2n+4$. To simplify matters, we also assume that the leftmost bottom joint is fixed both horizontally and vertically, and the rightmost bottom joint is fixed vertically. For the example truss in the figure, $n = 2$. Writing down the equations for each joint from 2 to $2n+4$, first the one corresponding to the horizontal force, then the vertical one, with $a = \frac{\sqrt{2}}{2}$, we get the following set of equations:



Joint 2, horiz: $af_1 - f_4 - af_5 = 0$
Joint 2, vert : $af_1 + f_3 + af_5 = 0$
Joint 3, horiz: $f_2 - f_6 = 0$
Joint 3, vert : $f_3 = -w_1$
Joint 4, horiz: $f_4 - f_8 = 0$
Joint 4, vert : $f_7 = 0$
Joint 5, horiz: $af_5 + f_6 - af_9 - f_{10} = 0$
Joint 5, vert : $af_5 + f_7 + af_9 = -w_2$
Joint 6, horiz: $f_8 + af_9 - af_{13} = 0$
Joint 6, vert : $af_9 + f_{11} + af_{13} = 0$
Joint 7, horiz: $f_{10} - f_{12} = 0$
Joint 7, vert : $f_{11} = -w_3$
Joint 8, horiz: $f_{12} + af_{13} = 0$

These are $N = 13$ equations for the same number of unknowns, f_i, $i = 1, \cdots, 13$. Note that every four equations correspond to the joints of one vertical beam. Note also that there are two types of vertical beams: those that have diagonal beams at the bottom joint (e.g. jount 5), and those that have diagonal beams at the top joint (e.g. joint 6). Thus, for non-boundary cases, every set of 8 consecutive equations has the same pattern as the next set of 8 equations.

Here is another truss for $n = 4$ and $N = 21$:



Generalizing for any even n, the set of 8 equations corresponding to 4 joints (e.g. joints 4, 5, 6, 7 in the second figure), we have, for $r = 5, 13, \cdots, 4n-3$ and $k = (r+3)/4$:

$$f_{r-1} - f_{r+3} = 0$$
$$f_{r+2} = 0$$
$$af_r + f_{r+1} - af_{r+4} - f_{r+5} = 0$$
$$af_r + f_{r+2} + af_{r+4} = -w_k$$
$$f_{r+3} + af_{r+4} - f_{r+7} - af_{r+8} = 0$$
$$af_{r+4} + f_{r+6} + af_{r+8} = 0$$
$$f_{r+5} - f_{r+9} = 0$$
$$f_{r+6} = -w_{k+1}$$

Note that the boundary joints' equations do not follow exactly the above general form. Also note that the matrix of the system is very sparse; it has at most 4 non-zero entries per row, independently of the size of n. Furthermore, although we can eliminate some equations and unknowns easily, in this assignment, we choose not to do so, in order not to complicate the pattern of the equations.

(a) [*20 points*] Write a MATLAB script which, for $n = 2, 4, 8, 16$, generates the matrix and right-hand side vector of the linear system, then solves the linear system (using backslash). For each n, the script also calculates and outputs the maximum and minimum forces over all members (note: the minimum may be negative), the maximum in *absolute value* over all *diagonal* members' forces, the maximum in *absolute value* over all *top horizontal* members' forces, as well as the condition number of the matrix.

Because the matrix A is sparse, we use sparse matrix techniques to generate it and store it. See example code below. You don't have to follow exactly the example code, but must have loops over n and over the matrix rows.
You can visualize the sparsity pattern of a sparse matrix A by `spy(A)`.
To get (an estimate of) the condition number of a sparse matrix A, use `condest`.
For the output of n, the four force quantities and the condition number, use
`fprintf('%3d %9.2f %9.2f %9.2f %9.2f %8.2f\n', ...)`

- For the case $n = 4$, also calculate the LU factorization (using the `lu` function in matlab) of A, and plot (using `spy`), the sparsity patterns of A, L, U and P. *You must keep an ordering of the equations and unknowns as in the example,* otherwise, the sparsity patterns will not be easy to study. Starting from the left and proceeding to the right, for the equations go: top horiz, top vertic, bot horiz, bot vertic, etc. Do not forget the last equation (rightmost joint, horizontal).
- For the case $n = 16$, plot, in one plot, four lines: the top horizontal members' forces (solid) versus their index, the vertical members' forces (dashed) versus their index, the diagonal members' forces (dashed-dotted) versus their index, and the bottom members' forces (dotted) versus their index.
- After the loop of n, the script plots the top horizontal members' forces (f_i, $i = 4, 8, \cdots, N - 5$) versus their normalized (by the respective N) index, in one plot (four lines plotted, $n = 2$ solid, $n = 4$ dashed, $n = 8$ dashed-dotted, $n = 16$ dotted).

Assuming all calculated forces are stored in matrix `t`, each column corresponding to a different n, to plot the top horizontal members' forces versus their normalized index use

```
plot([4:4:ni(1)-5]/ni(1), t(4:4:ni(1)-5, 1), 'r-', ...
     [4:4:ni(2)-5]/ni(2), t(4:4:ni(2)-5, 2), 'g--', ...
     [4:4:ni(3)-5]/ni(3), t(4:4:ni(3)-5, 3), 'b-.', ...
     [4:4:ni(4)-5]/ni(4), t(4:4:ni(4)-5, 4), 'k:');
```

Example code

```
for nn = 1:4
    n = 2^nn; % no. of top horizontal members
    p = [0.5:n+1]'/(n+1); W = 10000*p.*p.*(1-p)/(n+1); % weights
    N = 4*(n+1)+1; ni(nn) = N; % no. of unknowns
    A = sparse(N, N);
    b = zeros(N, 1); b(4:4:N-1, 1) = -W; % right-hand side vector
    a = sqrt(2)/2;
% leftmost vertical member
    A(1, 1:5) = [a  0  0 -1 -a];
    A(2, 1:5) = [a  0  1  0  a];
    A(3, 1:6) = [0  1  0  0  0 -1];
    A(4, 1:4) = [0  0  1  0];
    for k = 2:2:n-1 % go over pairs of interior vertical members
% vertical member with diagonal beams at the bottom joint
        r = 4*k-3;
        A(r,   r-1:r+3) = [1  0  0  0 -1];
        A(r+1, r+1:r+2) = [0  1];
        A(r+2, r:r+5)   = [a  1  0  0 -a -1];
        A(r+3, r:r+4)   = [a  0  1  0  a];
% vertical member with diagonal beams at the top joint
        fill-in the matrix entries
        sparsity patterns, plots, etc
    end
% two rightmost vertical members
fill-in the matrix entries
% rightmost joint
fill-in the matrix entries
solve system, output, plots, etc
```

(b) [*25 points*] What are the lower, upper and total bandwidths of A for any n? Explain. What is the (exact) number of nonzero entries in A in terms of n? Explain.
What can you say about the permutation matrix P (for any n) arising from the LU factorization with partial pivoting? Is there some ordering of the equations that would result in $P = \mathbb{I}$? If yes, describe the ordering and how you thought of it.
What are the lower, upper and total bandwidths of U for any n? Explain. What are the lower, upper and total bandwidths of L for any n?
Based on the numerical results, how does the condition number of A behave approximately in terms of n?
Based on the numerical results, how do the values of the top horizontal members' forces behave as we go from left to right? How do the maximum and minimum values depend on n (approximately)?
Based on the numerical results, how do the values of the diagonal members' forces behave as we go from left to right? How do the maximum and minimum values depend on n (approximately)?

Submit a printed hard-copy of your MATLAB script, output, plots, and your (hand-written or typed) comments and answers. In addition, submit your MATLAB script (.m file) electronically through the CDF system.

2. For (b) and (c), assume you have a computer system that uses two-decimal digits floating-point arithmetic with (proper) rounding. Let

$$A = \begin{bmatrix} 0.01 & 1 \\ -1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

(a) [*5 points*] Compute the LU factorization of A using Gauss elimination without pivoting and applying exact (fractional) arithmetic. Indicate the computed L and U factors. Show your calculations.
Using the computed L and U matrices and forward and back substitutions, compute the solution to $Ax = b$, again applying exact (fractional) arithmetic. Indicate the computed solution vector and any intermediate vector arising from the forward substitution. Show your calculations.

(b) [*9 points*] Compute the LU factorization of A using Gauss elimination without pivoting and applying two-decimals floating-point arithmetic to each operation. *Indicate* the computed L and U factors. Show your calculations.
Using the computed L and U matrices and forward and back substitutions, compute the solution to $Ax = b$, again applying two-decimals floating-point arithmetic to each operation. Indicate the computed solution vector and any intermediate vector arising from the forward substitution. Show your calculations.

(c) [*11 points*] Do the same as in (a), but with partial (row) pivoting, again applying two-decimals floating-point arithmetic to each operation. Indicate the permutation matrix P arising, and the computed L, U, x and any intermediate vector arising from the forward substitution. Show your calculations.

3. Assume that, when solving $Ax = b$ on the computer, A is represented as $\hat{A}$, while b does not have any representation error. Also assume there is no error arising from computations, i.e., the computed solution $\hat{x}$ satisfies $\hat{A}\hat{x} = b$.

(a) [*15 points*] Show that, if $\|A^{-1}\|\|A - \hat{A}\| < 1$, then

$$\frac{\|x - \hat{x}\|}{\|x\|} \le \frac{\kappa(A)}{1 - \|A^{-1}\|\|A - \hat{A}\|} \frac{\|A - \hat{A}\|}{\|A\|}$$

(b) [*15 points*] Show that, if $\delta = \|A^{-1}\|\|A - \hat{A}\| < 1$, then, within $O(\delta^2)$ accuracy,

$$\frac{\|x - \hat{x}\|}{\|x\|} \le \kappa(A) \frac{\|A - \hat{A}\|}{\|A\|}$$

Hints: At some point in (a), it may be helpful to write $\|\hat{x}\|$ as $\|(\hat{x} - x) + x\|$. In (b), you can assume you have shown (a); it may then be helpful to consider the Taylor series of $f(x) = \frac{1}{x}$, about $x = 1$, with stepsize δ.
Note: You cannot assume that the inequality

$$\frac{\|x - \hat{x}\|}{\|x\|} \le \kappa(A) \left(\frac{\|b - \hat{b}\|}{\|b\|} + \frac{\|A - \hat{A}\|}{\|A\|} + \frac{\|r\|}{\|b\|} \right)$$

holds. You are actually requested to prove a "part" of it.