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. Consider the matrix A and its inverse A^{-1},

$$A = \begin{bmatrix} 6 & 13 & -17 \\ 13 & 29 & -38 \\ -17 & -38 & 50 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 6 & -4 & -1 \\ -4 & 11 & 7 \\ -1 & 7 & 5 \end{bmatrix}.$$

(a) [*3 points*] What is the condition number of A in the infinity norm?

(b) [*6 points*] Suppose we solve $Ax = b$ for some b, and obtain $\hat{x}$, so that $\|b - A\hat{x}\|_\infty \le 0.01$. How small an upper bound can be found for the absolute error $\|x - \hat{x}\|_\infty$? Give the bound as a numerical value.

(c) [*6 points*] With the same situation as in (b), how small an upper bound can be found for the relative error $\dfrac{\|\hat{x} - x\|_\infty}{\|x\|_\infty}$? Give the bound in terms of $\|b\|_\infty$.

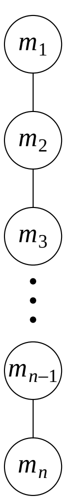
2. [*20 points*] Consider the linear system

$$0.00211x_1 + 0.08204x_2 = 0.04313$$

$$0.337x_1 + 12.84x_2 = 6.757$$

Solve the system using Gauss elimination and applying 4-decimal-digits floating-point arithmetic with proper rounding. The results of each operation (addition, multiplication, division) of GE must be stored using 4-decimal-digits floating-point representation. Do this three times: (a) without pivoting, (b) with partial pivoting, (c) with complete pivoting. Indicate the intermediate results (multipliers, upper triangular matrix), and $\hat{x}$ for each case.
In each of the three cases, what are the relative errors (in abs. value) for each component of $x = (x_1, x_2)^T$, and what is the relative error for x in the infinity norm? (That is, what are $\dfrac{|x_1 - \hat{x}_1|}{|x_1|}$, $\dfrac{|x_2 - \hat{x}_2|}{|x_2|}$, and $\dfrac{\|x - \hat{x}\|_\infty}{\|x\|_\infty}$?) Present the errors in table form (three cases as rows, and three errors as columns), and comment. Exact solution is $(1.000, 0.500)^T$.

3. A group of n parachutists each with given mass m_i and drag coefficient c_i, $i = 1, \cdots, n$, are connected by a weightless cord, and are falling at a given velocity v. We would like to calculate the tension t_i, $i = 1, \cdots, n-1$, in each section of the cord and the acceleration a of the whole group. Let's index the parachutists from top ($i = 1$) to bottom ($i = n$), and let $g = 9.81$ be the acceleration of gravity. For the top parachutist ($i = 1$), Newton's second law gives the equation



$$t_1 + m_1 g - c_1 v = m_1 a$$

- do not use any symbolic variables, packages, etc.
- write by hand the equations for n= 6
- use code given to generate A
- do help spdiags, help ones
- ask early enough

which can be written as

$$t_1 - m_1 a = c_1 v - m_1 g. \tag{1}$$

For an arbitrary "interior" parachutist indexed i, $i = 2, \ldots, n-1$, Newton's second law gives the equation

$$-t_{i-1} + t_i + m_i g - c_i v = m_i a$$

which can be written as

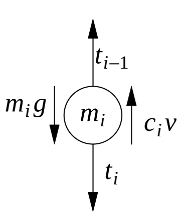
$$-t_{i-1} + t_i - m_i a = c_i v - m_i g \tag{2}$$

For the bottom parachutist ($i = n$), Newton's second law gives the equation

$$-t_{n-1} + m_n g - c_n v = m_n a$$

which can be written as

$$-t_{n-1} - m_n a = c_n v - m_n g \tag{3}$$



For convenience, let's denote the unknown a by t_n. Writing equations (1), (2) for $i = 2, \cdots, n-1$ and (3) (in that order), we get a linear system of equations $At = b$, with respect to the unknowns t_i, $i = 1, \cdots, n$. Note that the matrix of the system is very sparse; it has at most 3 non-zero entries per row, independently of the size of n. (However, it is not tridiagonal.)

generate matrix and vector

(a) [*20 points*] Write a MATLAB script which, for $n = 4, 8, 16, 32$, generates the matrix and right-hand side vector of the linear system, then solves the linear system (using backslash). After the loop of n, the script plots the tension vectors components (not including the acceleration), versus their normalized (by the respective n) index, in one plot (four lines plotted). Do this twice, (i) with $v = 6$, $m_i = 50 + 50\,\frac{i-1}{n-1}$, and $c_i = 25 - 10\,\frac{i-1}{n-1}$, and (ii) with $v = 6$, m_i drawn randomly in the interval $(50, 100)$, then sorted from smallest to largest, and c_i drawn randomly in the interval $(15, 25)$, then sorted from largest to smallest.

plots

twice (2 setups)

In the case (i), for each n, also get and output the condition number of the matrix, and the maximum tension computed. At the end of the loop for n, and for the case (i), plot in **log-log** scale (`loglog`) the condition numbers versus n, using a solid line and thick dots for the data (`'k.-'`).

case (i) also includes condition number and max tension

Based on the numerical results, comment on how the acceleration and the maximum tension behave with n. How do the components of the tension vectors vary with their index? Where (for which i) does the max tension occur? Also comment on how the condition numbers behave with n. Submit a hard-copy of your script, output, plot and comments.

Notes: For (i), use `m = linspace(50, 100, n)'` and `c = 25 - 10*linspace(0, 1, n)';` For (ii), use `m = sort(50 + 50*rand(n, 1), 'ascend');` and `c = sort(15 + 10*rand(n, 1), 'descend');`

Because the matrix A is sparse, we use sparse matrix techniques to generate it and store it. E.g `e = ones(n, 1); A = spdiags([-e, e], [-1, 0], n, n); A(:, n) = -m;` Note that 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`.

If you have four vectors of n_i, $i = 1, \cdots, 4$, components respectively, to plot their components versus their normalized index use

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

(b) [*13 points*] What would the forms of the L and U factors and of the permutation matrix P be, if LU factorization with row pivoting was applied to the matrix A? Your answer should be given in terms of n and m_i (summation notation is acceptable). Note that this is a mathematical question, but MATLAB could help you get ideas.

(c) [*12 points*] Find (mathematically) a closed form formula for the acceleration, and for the tensions t_i, $i = 1, \cdots, n-1$, in terms of n, m_i, c_i, v and g. Justify mathematically where (for which i) the maximum tension occurs.

4. [*20 points*] Assume that A and B are given dense $n \times n$ matrices, B is non-singular, $\mathbb{I}$ is the identity matrix of order n, and b a given $n \times 1$ vector, for some n large. Explain how you would efficiently compute $z = B^{-1}(2A + \mathbb{I})(B^{-1} + A)b$. Give, in terms of n, approximate operation counts for all computations you propose.

Note: The computations that you will propose may include LU factorization, back-and-forward substitutions, matrix-vector products, matrix-matrix products, matrix inverse calculation, addition of vectors or matrices, and other similar computations. However, you are **not** obliged to use **all** these types of computations. For each computation you propose, you should give operation counts (indicating the highest power of n, including the coefficient), and make sure that the total number of operation counts is as little as possible. You do **not** need to describe algorithms for these computations. Also note that, while B is given, B^{-1} is not given.