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

All questions **must** be written in latex, and compiled to a single pdf. Plots (3 for Q2 and 1 for Q3) must in embedded in latex. All code and output of Q2 and Q3 should be *embedded* into latex. Code and output should be embedded with `fixed-width fonts`, e.g. Courier. Font size of all fonts must be 12. Linespacing set to 1.1.
What to submit:
(1) The single pdf file 00A2.pdf (with embedded code and output).
(2) The latex source file of A2.
(3) Any source code you wrote (for computation, plots, etc).
(4) Plot files (7 for Q2 and 1 for Q3) in eps and png format.
(If `subplot` is used for the last 4 plots of Q2, then this gives only one file, so a total of 4 files for Q2.)
Thus, the code and plots will be available within latex/pdf, *as well as* separately.

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?   bound of relative and absolute error

(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.   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

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

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.)
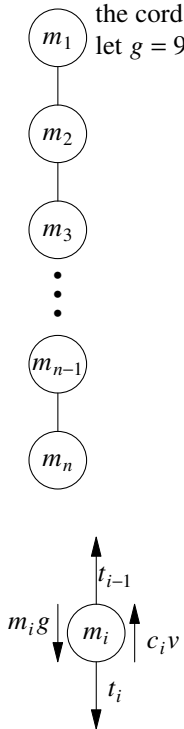
(a)   [*25 points*] Write a MATLAB (or equivalent) script which, for $n = 8, 16, 32, 64$, generates the matrix and right-hand side vector of the linear system, then solves the linear system (using backslash), and outputs n, the maximum and minimum tension, the acceleration and the condition number of A, using format `fprintf('%3d %10.3e %10.3e %10.3e %10.3e0, ....` 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 $v = 6$, $m_i = 50 + 50\,\dfrac{i-1}{n-1}$, and $c_i = 50 - 20\,\dfrac{i-1}{n-1}$, 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 $(30, 50)$, then sorted from largest to smallest. Use appropriate labels and a legend. In latex, the two plots must be placed side-to-side with captions and subcaptions. At the end of the loop for n, and for the case (i) only, plot in **log-log** scale (`loglog`) the condition numbers versus n, using a solid line and thick dots for the data (`'k.-'`). Use appropriate labels.

≡ Based on the numerical results (including plots), comment on how the acceleration and the maximum and minimum tensions 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.

Also outside the loop for n, and for the case (i) and $n = 16$ only, plot the sparsity patterns of A, P, L, U, in a 2×2 format, either using latex or using `subplot` in matlab. Use appropriate titles and caption. Comment about whether they agree with what you expected. (These comments will be elaborated further in (b).)

Notes: For (i), use `m = linspace(50, 100, n)'` and `c = 50 - 20*linspace(0, 1, n)'`; For (ii), use `m = sort(50 + 50*rand(n, 1), 'ascend')`; and `c = sort(30 + 20*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.

3. [*35 points*] The Hilbert matrix of order n is defined to have entries $(H_n)_{ij} = \dfrac{1}{i + j - 1}$, for $i = 1, \cdots, n$, $j = 1, \cdots, n$. The Hilbert matrices are known to be very ill-conditioned once n increases beyond a certain value. Use MATLAB (or equivalent) to carry out the following experiments.

(a) For $n = 2, \cdots, 13$ do the following:
 – construct the Hilbert matrix H_n; you may use the built-in MATLAB function hilb(n) or your own function;
 – construct the vector b_n, such that $(b_n)_i = \displaystyle\sum_{j=1}^{n} \dfrac{j}{i + j - 1}$, for $i = 1, \cdots, n$;
 – solve $H_n x_n = b_n$ by Gauss elimination; use the backslash \ operator to obtain the solution (help slash); let $\hat{x}_n$ be the computed solution; note that the exact solution is $x_n = (1, 2, 3, \cdots, n)^T$; let $r_n = b_n - H \hat{x}_n$ be the residual;
 – compute the condition number of H_n, the computational relative error $\|x_n - \hat{x}_n\|/\|x_n\|$, the residual $\|r_n\|$, and $\|b_n\|$, where $\| \cdot \|$ is the same norm as that used for the condition number; use the built-in MATLAB functions cond() for computing the condition number of H_n and norm() for computing the norms of vectors (or matrices); (note that both cond() and norm() use Euclidean norms); store the condition number of H_n in the nth component of a vector c, and the respective relative error, norm of residual and of right side vector in the nth component of vectors e, r, v, respectively.
 – output n and the four quantities computed above with format `fprintf('%2d %9.2e %9.2e %9.2e %9.2e\n', ...`
 – Comment on the output and possible warnings by matlab.

(b) After exiting from the above for-loop,
 – In one graph, plot, in semilogy scale, c (the condition number of H_n) versus n, e ($\|x_n - \hat{x}_n\|/\|x_n\|$) versus n, and $c(n)r(n)/v(n)$ versus n. (Thus n will be in the x axis, and will be in regular scale, and the y axis will be in log scale.) To distinguish between the three lines in the plot a solid one for the condition numbers, a dashed line for the relative errors, and a dotted line for the third line. Use appropriate labels and a legend.
 – Interpret what you observe in the graph, and discuss whether it agrees with the theory.

Note: You may assume that MATLAB uses double precision. In other language, you must use double precision.