Worth: 10% **Due:** Before 10:00pm on **Monday 26 October 2015**.

Your submission must be a PDF file named a2.pdf and it must be handed-in using the MarkUs system. You are to create the PDF file using a document preparation system (*e.g.* Word or LaTeX). Only hand in the PDF file - do not hand in other files (*e.g.* a .doc file or a .tex file). (If you expect to have a difficult time with the typesetting, speak with Tom.)

For questions where you are asked to write a program, you must include your program and any output in your submission.

See the Syllabus for the course policy on late assignment submissions.

---

1. Consider the linear system

$$\begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = 1, \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 = 0, \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = 0. \end{cases}$$

   (a) Solve the system using *exact* arithmetic using any correct method.

   (b) Put the system in matrix form using a two decimal-digit chopped representation. That is, use $F(10, 2, L, U)$ and chopping.

   (c) Solve the system in (b) using Gauss elimination and two decimal-digit chopped arithmetic.

   (d) Solve the system in (b) using Gauss elimination and exact arithmetic.

   (e) What observations can you make?

   On some elimination steps you may encounter the situation that a computed value is not exactly 0.0 even though the algorithm is designed to make the value 0. This is caused by round-off errors. **Set the eliminated element to 0.0** because that is the intent of the Gauss Elimination algorithm.

2. If $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are $n \times n$ matrices, with $\mathbf{B}$ and $\mathbf{C}$ nonsingular, and $\underline{w}$ is an $n-$vector, how would you evaluate the formula

$$\underline{x} = \mathbf{B}^{-1}(\mathbf{C}^{-1} + \mathbf{A})(2\mathbf{A} + \mathbf{I})\underline{w}$$

   to compute $\underline{x}$ without computing any matrix inverses?

   What is the operation count for your algorithm?

3. (a) Using the programming languange of your choice (**other than** `matlab`), write a function that implements Gaussian elimination with no pivoting, a function that implements Gaussian elimination with partial (row) pivoting and a function that implements Gaussian elimination with complete pivoting. Your functions should have parameters `A` and `b` and should return a vector `x`. The vector `x` should be a computed solution to the linear system `A x = b`. Your functions are not required to return an `LU` factorization or pivoting information. They need only return an approximate solution to the problem.

(b) Use a random number generator to generate several linear systems with random coefficient matrices. Choose the right-hand sides to be such that correct solution has $x_i = (-1)^{i+1}$. Compare the accuracy and residuals of each of your Gaussian elimination implementations by using your functions to solve the systems. Vary the system dimension as well as the coefficients. Report on your observations.

(c) Can you devise a (nonrandom) matrix for which complete pivoting is significantly more accurate than partial pivoting? Demonstrate. (**HINT:** Read Computer Problem 2.7 on page 102 of the Heath text.)

4. (a) Gaussian elimination with partial pivoting is used to solve a 2×2 system $\mathbf{A}\underline{x} = \underline{b}$ on a computer with machine epsilon 10^{-20}. It is known that $\kappa_\infty(\mathbf{A}) \approx 10^{12}$ and that the exact solution is given by

$$\underline{x} = \begin{pmatrix} 12.345678901234567890 \\ 0.001234567890123456 \end{pmatrix}.$$

Underline the digits in x_1 and x_2 that probably agree with the corresponding digits in the computed solution. Explain the heuristic assumptions used to answer this question.

(b) It is known that the components of the exact solution to a linear system $\mathbf{A}\underline{x} = \underline{b}$ range from 10^{-2} to 10^4 in size, and that $\kappa(\mathbf{A}) \approx 10^6$. What must the machine precision be in order to ensure that the smallest component of `x = A\b` has at least six significant digits of accuracy? No proof is necessary, just a reasonable heuristic argument.

5. In this question, I will ask you to write a short `matlab` program. I will give a brief introduction to `matlab` in class, but in the meantime, you can log in to CDF and start `matlab`. Enter the command `demo` in the command window. If you prefer to use a `matlab` clone, that might be okay, but please check with Tom.

Consider the dimension n linear system $\mathbf{A}\underline{x} = \underline{b}$, where the elements of $\mathbf{A}$ and $\underline{b}$ are given by

$$a_{i,j} = j^i, \qquad 1 \leqslant i, j \leqslant n, \qquad \text{and} \qquad b_i = \sum_{j=1}^{n} (-1)^{j+1} a_{i,j}, \qquad 1 \leqslant i \leqslant n.$$

The exact solution is $\underline{x}^\star = [1, -1, \ldots, (-1)^{n+1}]^T$. For example, if $n = 3$, we have

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 4 & 9 \\ 1 & 8 & 27 \end{bmatrix}, \qquad \underline{b} = \begin{bmatrix} 2 \\ 6 \\ 20 \end{bmatrix} \qquad \text{and} \qquad \underline{x}^\star = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}.$$

Using `matlab`, set up and then solve the system $\mathbf{A}\underline{x} = \underline{b}$ for dimensions $n = 1, 2, 3, \ldots$. Stop when the relative error in the computed solution is greater than 1. At this point the computed solution will have no accurate digits. Use the `matlab` \ operator to solve the $\mathbf{A}\underline{x} = \underline{b}$ systems. To verify that you are generating the correct coefficient matrix and right-hand-side vector, display and verify your results for $n = 3$.

While solving the system for different values of n, produce a table having the following headings:

n	relative error	relative residual	cond(A)	det(A)

The values in the relative error column will be the results for the different values of n of the computation of the relative error, $(\|\underline{x}^\star - \underline{x}\|/\|\underline{x}^\star\|)$. And the values in the relative residual column will be the results for the different values of n of the computation of relative residual, $(\|\underline{r}\|/\|\underline{b}\|)$, where $\underline{r} = \underline{b} - \mathbf{A}\underline{x}$. The values in the cond(A) and det(A) columns will be the condition number of $\mathbf{A}$ and the determinant of $\mathbf{A}$, respectively. You should observe that the relative residual always remains small, while the relative error grows.

We know that a matrix $\mathbf{M}$ is singular if and only if $\det(\mathbf{M}) = 0$, and that $\mathbf{M}$ is poorly conditioned if it is almost singular. Is the value of $|\det(\mathbf{M})|$ a good indicator of the conditioning of $\mathbf{M}$? Comment.

Use the `matlab` \ operator to solve linear systems of the form $\mathbf{M}\underline{s} = \underline{t}$. Use the `matlab` function `norm` to compute (2-)norms, and the `matlab` function `cond` to compute (2-norm) condition numbers. Use the `matlab diary` statement to save your execution output in a file. See, for example, the results from executing the `matlab` commands `help slash`, `help norm`, `help cond`, `help det` and `help diary`.

To keep your grader happy, produce an easy to read table. Use `sprintf` to display numbers nicely, and choose a format that shows all digits that you believe to be significant.

If you have questions or encounter any difficulties, post a description of the issue on the Piazza course forum.