CS 3430: S24: Scientific Computing
Midterm 01

Vladimir Kulyukin
Department of Computer Science
Utah State University

February 08, 2024

Instructions

1. This exam has 10 problems worth a total of 10 points. You may use your class notes, my lecture PDFs and code samples in Canvas, and your homework solutions. You may not use any other materials (digital or paper).
2. You will type and save your solutions in `cs3430_s24_midterm01.py` and submit this file (and, if necessary, other files – see item 7 below) **in Canvas by 11:59pm on Feb. 14, 2024**.
3. Write your name and A-number in `cs3430_s24_midterm01.py`.
4. You may not talk to anyone when you are working on this exam orally, digitally, or in writing. Telepathy, if detected, will be dealt with as needed. I am doing my best to improve my means of detecting it.
5. You may use your interactive Python IDE, including the Python documentation that comes with it.
6. You may use your solutions to the previous assignments. For example, I will have no problem with you doing imports from your previous solutions as follows

   ```
   from cs3430_s24_hw01 import leibnitz_det, cramers_rule
   ```

   and then using your implementation of these functions from Assignment 01 to solve a midterm problem.
7. This is important! **Remember to include into your submission zip all the Python files you are importing from.** Thus, if you are importing anything from your `cs_3430_s24_hw02.py`, include it in the zip. When I run unit tests on your submission, I will put all your files into the same working directory with your `cs3430_s24_midterm01.py`.
8. You may not use any third party libraries in this exam, except the ones we used, e.g., `numpy` and `matplotlib`. You may use **only** your own solutions to previous/current assignments; you may use the unit tests in `cs3430_s24_midterm01_uts.py` to test your solutions.
9. If you can, do me a favor and write below your name and A-number in `cs3430_s24_midterm01.py` how much time you spent on this exam. I give you my word that I won't make it public anywhere. This is only for me to assess the easiness/difficulty/reasonableness of the exam.
10. I wish you best of luck and, as always, Happy Hacking and Thinking!

Problem 01 (1 point)

Implement the function `solve_lin_sys_with_gje(A, b)` that uses Gauss-Jordan Elimination to return the vector $\mathbf{x}$, if it exists, that solves the linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{x}$ is an $n \times 1$ column vector, and $\mathbf{b}$ is also an $n \times 1$ column vector. Use your implementation to solve respresented as augmented matrices in the unit tests.

Problem 02 (1 point)

Implement the function `solve_lin_sys_with_cramer(A, b)` that uses Cramer's rule and Leibnitz's determinant formula to solve the linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an $n \times n$ matrix and $\mathbf{b}$ is an $n \times 1$ column vector. In other words, your implementation of this function must use Leibnitz's formula, not the diagonal pivot multiplication formula used in `np.lingalg.det()`. The function returns a $n \times 1$ column vector $\mathbf{x}$. Use your implementation to solve the linear systems specified as augmented matrices in the unit tests.

Problem 03 (1 point)

Implement the function `solve_lin_sys_with_bsubst(A, n, b, m)` that uses back substitution to solve the linear system $Ax = b_1, b_2, ..., b_m$, where A is an $n \times n$ upper-triangular matrix and b is an $n \times m$ matrix of m $n \times 1$ column vectors b_1, b_2, ..., b_m. This function returns the $n \times m$ matrix X of m $n \times 1$ vectors x_1, x_2, ..., x_m such that $Ax_1 = b_1$, $Ax_2 = b_2$, ..., $Ax_m = b_m$. Use your implementation to solve the linear systems in the unit tests.

Problem 04 (1 point)

Implement the function `solve_lin_sys_with_fsubst(A, n, b, m)` that uses forward substitution to solve $Ax = b_1, b_2, ..., b_m$, where A is an $n \times n$ lower-triangular matrix and b is an $n \times m$ matrix of m $n \times 1$ column vectors b_1, b_2, ..., b_m. This function returns the $n \times m$ matrix X of m $n \times 1$ vectors x_1, x_2, ..., x_m such that $Ax_1 = b_1$, $Ax_2 = b_2$, ..., $Ax_m = b_m$. Use your implementation to solve the linear systems in the unit tests.

Problem 05 (1 point)

Implement the function `solve_lin_sys_with_lud(A, n, b, m)` to solve the linear system $Ax = b_1, b_2, ..., b_m$, where a is an $n \times n$ matrix, b is an $n \times m$ matrix of m $n \times 1$ vectors $b_1, b_2, ..., b_m$. Use your implementation to solve the linear systems in the unit tests.

Problem 06 (1 point)

Define the following concepts and answer question 8. Be brief and to the point. Do not write essays. A few sentences per item suffice.

1. Standard Maximization Problem (SMP);
2. Objective Function;
3. Corner Point;
4. Feasible Set;
5. Two conditions when the simplex algorithm stops;
6. Bounded Feasible Set;
7. Unbounded Feasible Set;
8. Does the simplex algorithm work on any linear programmng problems or only on SMPs?

Problem 07 (1 point)

Consider the following SMP. Maximize $p = 13x + 7y + 5z$ satisfying the following constraints:

1. $x \geq 0$;
2. $y \geq 0$;
3. $z \geq 0$;
4. $6x + z \leq 122$;

5. $2y + 5z \leq 502$;
6. $9x - 7y + 6z \leq 902$.

1. Write the slack equations for this problem. Clear specify each slack variable;
2. Set up the initial tableau for this SMP.

Problem 08 (1 point)

Find the pivot in the following simplex tableau.

	x_0	x_1	x_2	x_3	x_4	x_5	B.S.
x_3	6	6	7	1	0	0	190
x_4	12	7	22	0	1	0	510
x_5	22	10	12	0	0	1	810
p	-7	-22	-12	0	0	0	0

Problem 09 (1 point)

What is numerical instability? Give an example and a brief explanation. Do not write an essay. A paragraph will do.

Problem 10 (1 point)

A farmer has 1,000 acres and water rights to 600 acre-feet of water for next season. An acre-foot of water is the amount of water which covers 1 acre at a depth of 1 foot. Crop A yields 120 bushels per acre and requires 6 inches of water per acre, crop B yields 80 bushels per acre and requires 4 inches of water per acre, and crop C yields 50 bushels per acre and requires 4 inches of water per acre. The farmer expects crop A to yield a profit of $1.00 per bushel, crop B a profit of $1.20 per bushel, and crop C a profit of $2.00 per bushel. Solve for the land allocation for the farmer to grow the three crops for maximum profit.

Use the following decision variables: x_0 for the number of acres for crop; x_1 for the number of acres for crop B; and x_2 for the number of acres for crop C.

What to Submit

Submit your solutions in `cs3430_s24_midterm01.py`. **Remember to include into your submission zip all the Python files you're importing from.** In other words, if you are importing anything from your `cs_3430_s24_hw02.py`, include it in the zip. When I run the unit tests on your submission, I will put all your files into the same working directory with your `cs3430_s24_midterm01.py`