CS 3430: S22: Scientific Computing
Midterm 01

Vladimir Kulyukin
Department of Computer Science
Utah State University

February 10, 2022

Instructions

1. This exam has 8 problems worth a total of 10 points; you may use your class notes and my lecture pdfs in Canvas; you may not use any other materials (digital or paper).
2. You will type and save your solutions in `cs3430_s22_midterm01.py` and submit this file (and, if necessary, other files – see item 7 below) **in Canvas by 11:59pm Feb. 11, 2022**.
3. Write your name and A-number in `cs3430_s22_midterm01.py`.
4. You may not talk to anyone when you're working on this exam orally, digitally, or in writing. Telepathy, if detected, will also be dealt with.
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'll have no problem with you doing imports from your previous solutions as follows

   ```
   from cs3430_s22_hw01 import gje
   ```

   and then using your implementation of `gje` from Assignment 01 to solve a midterm problem.
7. This is important! **Remember to include into your submission zip all the Python files you're importing from.** Thus, if you are importing anything from your `cs_3430_s22_hw02.py`, include it in the zip. When we run unit tests on your submission, we'll put all your files into the same working directory with your `cs3430_s22_midterm01.py`. If you don't want to import from your files, you can copy and paste all your code into `cs3430_s22_midterm01.py`. Bottom line is that we need all your source code necessary and sufficient to run the unit tests.
8. You may not use any third party libraries in this exam. You may use **only** your own solutions to previous/current assignments; you may use the unit tests in `cs3430_s22_midterm01_uts.py` to test your solutions; I'll write other unit tests (very similar to the unit tests in `cs3430_s22_midterm01_uts.py`) to grade your submissions.
9. If you can, do me a favor and write below your name and A-number in `cs3430_s22_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_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 following two linear systems where the left matrix is $\mathbf{A}$ and the right matrix is $\mathbf{b}$.

$$\begin{pmatrix} 4 & 7 & -10 \\ 0 & 25 & 61 \\ 0 & 0 & -32 \end{pmatrix} \begin{pmatrix} -6 & 15 & 31 \\ 11 & 14 & -50 \\ -54 & 39 & 114 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & -1 & 11 \\ 0 & 2 & 6 & 12 \\ 0 & 0 & -15 & 13 \\ 0 & 0 & 0 & 35 \end{pmatrix} \begin{pmatrix} -5 & 22 & 23.5 & 798.45 & 1 & 0 & 11 \\ 112 & 49 & 12.25 & 69.43 & 1 & 0 & 12 \\ -63 & 20 & 356.78 & -34.97 & 1 & 0 & 13 \\ 51 & 13 & -12 & 10 & 1 & 0 & 14 \end{pmatrix}$$

Problem 02 (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 following two linear systems represented as augmented matrices.

$$\left[\begin{array}{ccc|ccc} 11 & 0 & 0 & -6 & 11 & 1 \\ 202 & 21 & 0 & 8 & 123 & 1 \\ -125 & 34 & 35 & 12 & 217 & 1 \end{array}\right]$$

$$\left[\begin{array}{cccc|cccc} 123 & 0 & 0 & 0 & -41 & 101 & 202 & 1000 \\ 256 & 217 & 0 & 0 & 22 & 123 & 491 & 786 \\ -17 & 303 & 168 & 0 & 4 & 21 & 7123 & 473 \\ 890 & 456 & 2789 & 13 & 567 & 3 & 87 & 257 \end{array}\right]$$

Problem 03 (2 points)

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 the following two linear systems respresented as augmented matrices.

$$\left[\begin{array}{ccc|c} 21 & 31 & -301 & -52 \\ 4 & 7 & -8 & 710 \\ 40 & 51 & -25 & 11 \end{array}\right]$$

$$\left[\begin{array}{cccc|c} 7 & -5 & 10 & 100 & 42 \\ 12 & 3 & 6 & 36 & 54 \\ -14 & 13 & 44 & 20 & 66 \\ 126 & 37 & 80 & 11 & 130 \end{array}\right]$$

Problem 04 (2 points)

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 $b_1, b_2, ..., b_m$. The function uses LU decomposition to factor the matrix `A` into U and L. Then it uses forward substitution to solve $Ly = b$ for y, uses back substitution to solve $Ux = y$ for x, and returns X, which is an $n \times m$ matrix of m $n \times 1$ vectors x_i such that $Ax_1 = b_1$, $Ax_2 = b_2$, ..., $Ax_m = b_m$. You may use your solutions to Problems 1 and 2 to solve this problem. Use your implementation to solve the following linear systems defined as augmented matrices where the left matrix represents $\mathbf{A}$ and the right matrix represents $\mathbf{b}$.

$$\begin{bmatrix} 173 & 2136 & 3173 & 4112 \\ 561 & 6165 & 7146 & 814 \\ 6137 & 743 & 8183 & 973 \\ 5196 & 940 & 7144 & 931 \end{bmatrix} \begin{bmatrix} 54.0 & 11.0 \\ -12.0 & 25.0 \\ 35.0 & 37.0 \\ 52.0 & 48.0 \end{bmatrix}$$

$$\begin{bmatrix} 737 & 1365 & 8173 & 9112 & 89.0 \\ 761. & 7165. & 7146. & 9014. & 765.0 \\ 3137. & 243. & 4183. & 573. & 876.0 \\ 1965. & 340. & 5144. & 831. & 1234.0 \\ 87. & 65. & 21. & 234. & 897. \end{bmatrix} \begin{bmatrix} 14.0 & 11.0 & 20.0 \\ -17.0 & 24.0 & 151.0 \\ 389.0 & 34.0 & 142.0 \\ 523.0 & 14.0 & 153.0 \\ 389.0 & 141.0 & 2531.0 \end{bmatrix}$$

Problem 05 (1 point)

Implement the function `solve_lin_sys_with_cramer(A, b)` that uses Cramer's rule 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. The function returns a $n \times 1$ column vector $\mathbf{x}$. Use your implementation to solve the following linear systems specified as augmented matrices.

$$\begin{bmatrix} 0 & 13 & -37 & | & -17 \\ 24 & 36 & -13 & | & 74 \\ 42 & 52 & -23 & | & 103 \end{bmatrix}$$

$$\begin{bmatrix} 737. & 1365. & 8173. & 9112. & 89.0 & | & 14.0 \\ 761. & 7165. & 7146. & 9014. & 765.0 & | & -17.0 \\ 3137. & 243. & 4183. & 573. & 876.0 & | & 389.0 \\ 1965. & 340. & 5144. & 831. & 1234.0 & | & 523.0 \\ 87. & 65. & 21. & 234. & 897. & | & 389.0 \end{bmatrix}$$

Problem 06 (1 point)

Define the following concepts:

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.

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

What to Submit

Submit your solutions in `cs3430_s22_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_s22_hw02.py`, include it in the zip. When we run unit tests on your submission, we'll put all your files into the same working directory with your `cs3430_s22_midterm01.py`