CS2810 Mathematics of Data Models	Gold
Spring 2019	February 20, 2019

Homework 05

Due: February 17, 9PM
Point total: 60

Instructions:

- Submit your PDF and/or .py file to Blackboard by the due date and time. Please do not zip your files together, as this interferes with Blackboard's preview functionality. Always show all your work, and for full credit, you must use the method that the problem instructs you to use (unless none is mentioned). Handwritten or typeset solutions are both acceptable, but unreadable submissions will be penalized. You may discuss problems with other students, but you may not write up solutions together, copy solutions from a common whiteboard, or otherwise share your written work or code. Do not use code or language that is copied from the Internet or other students; attribute the ideas *and* rephrase in your own words.

Problem 1 (15 points)

Given the linear recurrence $T(N) = 2T(N-1) + 3T(N-2)$, convert this recurrence to matrix form, find the eigenvalues of the matrix, and determine the asymptotic growth rate of the recurrence.

Problem 2 (15 points)

Find the diagonalization $A = PDP^{-1}$ of this matrix, then use the diagonalization to find its tenth power. $A = \begin{bmatrix} -1 & 1.5 \\ -4 & 4 \end{bmatrix}$

Problem 3 (10 points)

For each matrix, find its condition number using both the ℓ_1 and ℓ_∞ norms. Then state whether it is the more well-conditioned or the more ill-conditioned of the two.

i. $\begin{bmatrix} 1 & 2 \\ 1 & 2.1 \end{bmatrix}$

ii. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Problem 4 (10 points)

Show that if two matrices are similar to the same diagonal matrix, they are also similar to each other. (You may wish to use the fact that $(AB)^{-1} = B^{-1}A^{-1}$.)

Problem 5 (10 points)

Condition numbers might not be so bad if you have a fair amount of precision in your floating-point numbers to start. Here, you'll see how condition number interacts with Python's 64-bit floating point numbers.

Define a matrix $A = \begin{bmatrix} 1 & 1 + 1 \times 10^{-8} \\ 1 & 1 - 1 \times 10^{-8} \end{bmatrix}$

Using `numpy.linalg.cond()`, find the condition number of this matrix. Then multiply by $x = \begin{bmatrix} 100 \\ 200 \end{bmatrix}$

to get a vector $\vec{v}$, and finally, solve for $\vec{x_0}$ in $Ax_0 = \vec{v}$ by applying the inverse of A (`numpy.linalg.inv()`). How different is x_0 from x? If 64-bit floating point numbers generally have about 16 digits of precision, do these results support the rule that you lose about k digits of precision, if the condition number is 10^k? (Report the condition number, x_0, and the answers to the questions.)