

CSci 2033, S'18

Homework # 2

Due Date: 03/02/2018

Note: *This is an early draft directed at those who wish to start early (recommended!). Please check for updates.*

1. Consider in  $\mathbb{R}^2$ , the points satisfying the equation  $2x_2 - x_1 - 2 = 0$ . Show on a plot the points satisfying this equation. How are these points transformed by the matrix

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}?$$

2. What matrix rotates every point in  $\mathbb{R}^2$  by  $+\pi/6$  (i.e.,  $+30$  deg.)? [Explain how you get the general formula first. Hint: see how each of the basis vectors are transformed]
3. (a) If the system  $Ax = b$  has infinitely many solutions, is it possible for the system  $Ax = b'$  (same  $A$ , different right-hand side) to have only one solution? Could it have no solution? Explain.
- (b) If the system  $Ax = b$  is consistent and the reduced echelon form of  $A$  has a zero row, does the system  $Ax = b'$  (same  $A$ , different right-hand side) always have a solution? If your answer is yes explain, if not say what must be satisfied by  $b'$  in order for the system to have a solution.
4. (Magic squares). Associate a variable with each of the 9 cells of a  $3 \times 3$  array as follows:

$x_1$	$x_2$	$x_3$
$x_4$	$x_5$	$x_6$
$x_7$	$x_8$	$x_9$

It is required to find values for all the variables satisfying the magical property that the sum of the variables in every row, column, and the two main diagonals is the same, here 15. Write the resulting system of linear equations. Find the general solution to this system. You may use `rref` from matlab to perform the calculations.

Show that \*any\* solution has the property that  $x_i + x_j = 10$  when the indices  $i$  and  $j$  sum up to 10. Show one particular solution for which all values are nonnegative integers and which is different from the trivial solution of all 5's.

5. (matlab) In this exercise you will develop a script to determine the *standard* row-echelon form of a matrix. Your starting point will be the code `gaussp.m` seen in class earlier in the semester (posted). As was seen earlier in class, the only differences with `gaussp.m` script are: 1) No need for a right-hand side  $b$ ; 2) The row and column dimensions are now different ( $m$  rows,  $n$  columns); 3) the row number (called  $k$  in `gaussp`) evolves differently from the column number  $l$ . You will need to test if a sub-column is zero and if it is zero (or very close to zero) your algorithm needs to skip to the next column (i.e., increment  $l$  by one). If all subcolumns are zero then stop (completion). Some help will be provided in class.

```
if (MaxColEntry < tolA) ... 'skip to next column'
```

here `MaxColEntry` is the maximum absolute value of entries in the subcolumn being considered, and `tolA` is the tolerance `max(max(abs(A)))*1.e-12` computed at the very beginning of the script.

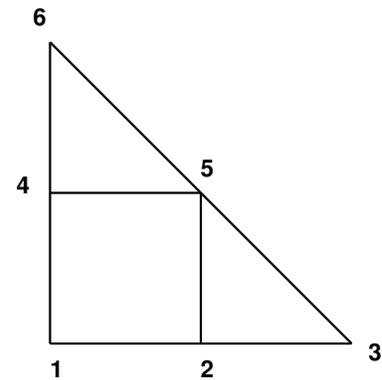
Your `rechln.m` script should also return an array `jb` which contains the indices of column pivots - just like `rref` does.

You need to run your script on the matrix called `A1` provided in `MathHw2.mat` save file (available from class home page – read with the command `load('MathHw2')`).

Hand in: print out of your row echelon script `rechln.m` and the diary of the execution for the data set provided.

6. A particle moves in a small triangular grid of 6 points with certain probabilities of moving from one point to another given by the following matrix:

	1	2	3	4	5	6
1	0	1/2	0	1/2	0	0
2	1/2	0	1	0	1/2	0
3	0	1/4	0	0	0	0
4	1/2	0	0	0	1/2	1
5	0	1/4	0	1/4	0	0
6	0	0	0	1/4	0	0



The table is to be read column-wise: if a particle is node  $j$  then in the next step it will be in columns 1, 2, ..., 6 with probabilities  $a_{1j}, a_{2j}, \dots, a_{6j}$ . Assume the particle is in node 1 initially. Do a few iterations (moves) to determine the particle be in nodes 1, 2, ..., 6 after 1, 2, 4 steps. What is the probability that the particle be at nodes 1, 2, ..., 6 after a very long time?

**Extra credit** (5pts) Find these probabilities by solving a homogeneous system.

7. (Matlab) First read the added section in Lecture notes # 6 on Google PageRank. We consider a small set of 80 web-pages. The links of the pages pointing to each other is captured by a matrix  $L$  that is such that

$$l_{ij} = \begin{cases} 1 & \text{if } j \text{ points to } i \\ 0 & \text{otherwise} \end{cases}$$

Because most of the entries of the matrix are zeros we say that the matrix is 'sparse' and matlab stores these matrices in a special format. One such matrix is saved in the matlab file `H2PRmat.mat` which can be read with the command `load('H2PRmat')`; – that will read a matrix called  $L$ . To see the pattern (nonzero entries represented by a dot) type `spy(L)`.

Your task is to run the PageRank algorithm in the notes and to determine the 5 top pages out of the 80. You will first need to create the Hyperlink matrix from  $L$ . This will amount to scaling each column by its column-sum (sum of all entries in the column). Then you will run 200 steps of the algorithm starting  $v$  with the vector of all ones. Use  $\delta = 0.9$  as in the class example.

**Extra credit** (5pts) Find out how to obtain the limit by solving a linear system of equations by doing something similar to what we did for the example of populations. Then use Gaussian elimination or Gauss-Jordan elimination [use posted scripts] to solve the system. NOTE: you may have to convert the matrix from sparse to standard format.. [Hint: explore the command `'full'...`]