

## Lab Session 1

MA-423 : Matrix Computations Lab

2018

R. Alam

1. The purpose of this exercise is to illustrate anomaly in automatic computation. On my computer, Matlab produces

$$\begin{aligned} \left(\frac{4}{3} - 1\right) * 3 - 1 &= -2.2204 \times 10^{-16} \\ 5 \times \frac{(1 + \exp(-50)) - 1}{(1 + \exp(-50)) - 1} &= \mathbf{NaN} \\ \frac{\log(\exp(750))}{100} &= \mathbf{Inf} \end{aligned}$$

Try on your machine. Can you explain the reason behind these anomalies?

1. Let  $A$  be a random matrix generated by `rand(8)`. Find the maximum values (a) in each column, (b) in each row, and (c) overall. Also use `find` to find the row and column indices of all elements that are larger than 0.25.
2. A *magic square* is an  $n$ -by- $n$  matrix in which each integer  $1, 2, \dots, n^2$  appears once and for which all the row, column, and diagonal sums are identical. MATLAB has a command `magic` that returns magic squares. Check its output for a few values of  $n$  and use MATLAB to verify the summation property. (The antidiagonal sum will be the trickiest. Look for help on how to “flip” a matrix.)
3. Consider the magic square  $A = \text{magic}(n)$  for  $n = 3, 4$ , or  $5$ . What does

`p = randperm(n); q = randperm(n); A = A(p,q);`

do to

`sum(A), sum(A')', sum(diag(A)), sum(diag(flipud(A))), rank(A)`

The magic square  $A = \text{magic}(4)$  is singular. What do

`null(A), null(A,'r'), null(sym(A)), and rref(A)`

tell you about linear dependence of columns of  $A$ ?

4. Are the following true or false? Assume  $A$  is a generic  $n$ -by- $n$  matrix.  
(a)  $A^{(-1)}$  equals  $1/A$  (b)  $A.^{(-1)}$  equals  $1./A$
5. Suppose  $p$  is a row vector of polynomial coefficients. What does this line do?

`(length(p)-1:-1:0) .* p`

6. (a) Look up `diag` in the online help and use it (more than once) to build the 16-by-16 matrix

$$D = \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix}$$

- (b) Now read about `toeplitz` and use it to build  $D$ .  
 (c) Use `toeplitz` and whatever else you need to build

$$\begin{bmatrix} 1 & 2 & 3 & \cdots & 8 \\ 0 & 1 & 2 & \cdots & 7 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 2 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{8} \\ \frac{1}{2} & 1 & \frac{1}{2} & \cdots & \frac{1}{7} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{1}{7} & \frac{1}{6} & \ddots & 1 & \frac{1}{2} \\ \frac{1}{8} & \frac{1}{7} & \cdots & \frac{1}{2} & 1 \end{bmatrix}$$

The second case looks best in format `rat`.

7. (a) Suppose  $A$  is a matrix whose entries are all positive numbers. Write one line expression that will multiply each column of  $A$  by a scalar so that, in the resulting matrix, every column sums to 1.  
 (b) Try this more difficult variation: Suppose that  $A$  may have zero entries, and leave a column of  $A$  that sums to zero unchanged.
8. A random Fibonacci sequence is generated by choosing  $x_1$  and  $x_2$  and setting  $x_{n+1} := x_n \pm x_{n-1}$ ,  $n \geq 2$ . Here  $+$  and  $-$  must have equal probability of being chosen. It is known that, with probability 1, for large  $n$  the quantity  $|x_n|$  is of order  $c^n$ , that is,  $|x_n| = \mathcal{O}(c^n)$ , where  $c := 1.13198824\dots$ . Your task is to test this assertion. Try the following script.

```
>> clear
>> rand('state', 1000)
>> x = [1, 2];
>> for n=2:999, x(n+1) = x(n)+sign( rand-0.5)*x(n-1); end
>> semilogy (1:1000, abs(x))
>> c =1.13198824;
>> hold on
>> semilogy(1:1000, c.^ [1:1000])
hold off
```

Try to understand what the above script does and why it does so. Use matlab command `help` to understand an in-built function/command whenever necessary.

\*\*\* End \*\*\*