

FF505 – Computational Science

Week 3, Spring 2016

Before you approach these exercises you are expected to have worked at the exercises from the slides. In particular, the creation and manipulation of a file in Linux, the creation of a specific matrix and the creation, saving and execution of script.

Work in small groups (eg, two persons) at the computer to solve the following exercises.

Exercise 1 Solving systems of linear equations

Use Matlab to solve the two systems of linear equations derived in class for the examples on electrical circuits and chemical equations:

$$\begin{aligned}i_1 - i_2 + i_3 &= 0 \\ -i_1 + i_2 - i_3 &= 0 \\ 4i_1 + 2i_2 &= 8 \\ 2i_2 + 5i_3 &= 9\end{aligned}\qquad\qquad\qquad\begin{aligned}x_1 &= 6x_4 \\ 2x_1 + x_2 &= 2x_3 + 6x_4 \\ 2x_2 &= 12x_4\end{aligned}$$

Exercise 2 Matrix operations

Use MATLAB to generate random 4×4 matrices $A = \mathbf{rand}(4)$ and $B = \mathbf{rand}(4)$. For each of the following, compute $A1$, $A2$, $A3$, and $A4$ as indicated and determine which of the matrices are equal (you can use MATLAB to test whether two matrices are equal by computing their difference):

1. $A1 = A \cdot B$, $A2 = B \cdot A$, $A3 = (A' \cdot B)'$, $A4 = (B' \cdot A)'$
2. $A1 = A' \cdot B'$, $A2 = (A \cdot B)'$, $A3 = B' \cdot A'$, $A4 = (B \cdot A)'$
3. $A1 = \mathbf{inv}(A \cdot B)$, $A2 = \mathbf{inv}(A) \cdot \mathbf{inv}(B)$, $A3 = \mathbf{inv}(B \cdot A)$, $A4 = \mathbf{inv}(B) \cdot \mathbf{inv}(A)$
4. $A1 = (\mathbf{inv}((A \cdot B)'))'$, $A2 = \mathbf{inv}(A' \cdot B')$, $A3 = \mathbf{inv}(A') \cdot \mathbf{inv}(B')$,
 $A4 = (\mathbf{inv}(A) \cdot \mathbf{inv}(B))'$

Compare the results with the ones you would instead achieve using the Matlab-specific, array operations, such as `.*`.

Exercise 3 An application of matrix calculus

Car owners occasionally trade in their used car for a new car and marketing people are interested in the following type of question: Assume that you own a Citroen. Will your next car be another Citroen or a Volkswagen? Customers' choices eventually determine the market share of different brands. Car dealers need estimates of how the market share of their brand (or brands) will change as a function of time. The problem can be dealt with by matrix algebra, and ends up being an exercise in matrix multiplication.

Let the index i , $1 \leq i \leq n$ denote car brands, in alphabetic order. That is, 1 is Alfa Romeo, 2 is Aston Martin, 3 is Bentley, and n is Toyota. Let also $t = 1, 2, \dots$ denote time, measured in

Volkswagen	Fiat	Ford	Peugeot	Toyota
426	436	364	437	336

Table 1: Cars traded at time 0 for each car brand.

new car (<i>i</i>)	current car (<i>j</i>)				
	Volkswagen	Fiat	Ford	Peugeot	Toyota
Volkswagen	335	717	586	340	104
Fiat	375	257	409	551	626
Ford	491	43	614	292	445
Peugeot	246	383	373	567	649
Toyota	554	600	18	250	177

Table 2: Cars traded from a brand to another.

years from an arbitrary initial time, and let $F_j(t)$ be the fraction of cars of type j traded in year t . Assume for simplicity that car owners trade their cars in every year¹ and let C_{ij} be the fraction of cars of brand j that are traded in for a new car of brand i .

As an example you will use the following data from Table 1 and 2 that are also available in electronic form from this xls file or this txt file². (We limit ourselves to $n = 5$ because the matrices we then have are easily visualized on a screen, there are of course more car brands). The data refer to the absolute numbers, divided by 1000, from which fractions can be derived. We will assume that the distribution of the next outcome depends only on the previous outcome and that the trading fractions are constant over time.

- Using the definition of C_{ij} , argue that $0 \leq C_{ij} \leq 1$ and that $\sum_{i=1}^n C_{ij} = 1$ for all j . In other words, all elements lie between zero and one, and the sum of all elements of any column of the matrix \mathbf{C} is equal to one.
- Write down the formula to calculate the fraction of cars for each brand after $t = 10$ time periods.
- Show that, with the assumptions taken that all current cars trade a new car, the number of cars distributed among brands is always 1, i.e.,

$$\sum_{i=1}^n F_i(t) = 1. \quad (1)$$

- Write a matlab program performing the following operations:
 1. Calculate $\mathbf{F}(t)$ according to formulas you have given in the point above
 2. Plot, as a function of time and for selected j , the market share $F_j(t)$ of car brand j .

Repeat by initializing \mathbf{C} as you prefer, but keeping in mind the above discussed restrictions on its elements and choosing any initial distribution $\mathbf{F}(0)$.

- You should ask yourself, and possibly answer, several questions. For example, does the market share of different brands approach a limit for large t , and does this limit depend on the initial condition?

¹This is common for fleet owners, but if you think trading a car in every year is a huge waste of money, just change the unit of time to e.g. five years.

²Tip: In MATLAB the command `xlsread` can be used to read files in the xls format; the command `load` to read data in plain text.

Exercise 4 Efficiency

Set $n = 200$ and generate an $n \times n$ matrix and two vectors in \mathbb{R}^n , both having integer entries, by setting

$$\begin{aligned} A &= \mathbf{floor}(10 * \mathbf{rand}(n)); \\ b &= \mathbf{sum}(A)'; \\ z &= \mathbf{ones}(n, 1) \end{aligned}$$

1. The exact solution of the system should be the vector z . Why?. Explain. One could compute the solution in MATLAB using the “\” operation or by computing A^{-1} and then multiplying A^{-1} times b . Let us compare these two computational methods for both speed and accuracy. One can use MATLAB’s **tic** and **toc** commands to measure the elapsed time for each computation. To do this, use the commands

$$\begin{aligned} \mathbf{tic}, x &= A \backslash b; \mathbf{toc} \\ \mathbf{tic}, y &= \mathbf{inv}(A) * b; \mathbf{toc} \end{aligned}$$

Which method is faster?

To compare the accuracy of the two methods, we can measure how close the computed solutions x and y are to the exact solution z . Do this with the commands

$$\begin{aligned} \mathbf{max}(\mathbf{abs}(x - z)) \\ \mathbf{max}(\mathbf{abs}(y - z)) \end{aligned}$$

Which method produces the most accurate solution?

2. Repeat part (a), using $n = 500$ and $n = 1000$.

Exercise 5 Magic squares

Read the documentation about magic squares `doc magic`. Are magic squares singular? Use the MATLAB command `det(magic(n))` to compute the determinants of the magic squares matrices in the cases $n = 3, 4, \dots, 10$. What seems to be happening? Check the cases $n = 24$ and $n = 25$ to see if the pattern always holds.