Homework 1 - Due Wed. Sep. 2nd
Math 561

Dr. Mauro Maggioni
Office: 293 Physics Bldg.
Phone: 660-2825
Web page: www.math.duke.edu/~ mauro
E-mail: mauro.maggioni *at* duke.edu

Homework Policies

Homework is due at the beginning of class, stapled, written legibly, on one side of each page only. Otherwise, it will be returned ungraded.
The logic of a proof (when proving a statement is part of the exercise) must be clear, with all steps justified. The clarity and completeness of your arguments will count as much as their correctness. Some problems from the homework will reappear on exams. The lowest homework score will be dropped. No late homework will be accepted. Duke policies apply with no exceptions to cases of incapacitating short-term illness, or for officially recognized religious holiday. You may, and are encouraged to, discuss issues raised by the class or the homework problems with your fellow students and both offer and receive advice. However all submitted homework must be written up individually without consulting anyone else's written solution.

The submission of homework that require numerical work on a computer should include the following: printout of the code used to solve the problems, of its inputs and of its outputs. The code should be written clearly, and should be commented at least in such a way that the input/outputs of the code is clear. The specific outputs requested by the exercise should be discussed in your writeup as needed in order to answer the questions in the problems. For example if the problem asks you to compare the results of two algorithms for solving a given linear system, you should exhibit the the code for the two algorithms, commented, the input to the algorithms and the two outputs, and comment on whether the results are the same or not, why, etc...
Finally, an e-mail with subject *FirstnameInitial_Secondname_homeworkMath561_xx*, where *xx* stands for the homework number, and attachment named *FirstnameInitial_Secondname_homeworkMath561_xx.zip*, should be e-mailed to me (before class). I may miss messages in my mailbox that do not adhere to this format. The .zip file should contain all the code needed to reproduce the results of the homework, printouts (in pdf). You may turn in the exercises either in writing or electronically. The filenames for the code should be indicative of the exercise, so for example if it is a Matlab script for exercise 3.14, it could be named Ex3_14.m .

If you have no access to a suitable computing environment, let me know: an account can be created for you in math, so that you will have physical and remote access to machines in the math department, on which Matlab is installed, for running code. I will need your Duke ID in order to submit such request.

Assignment

Review your linear algebra! Lectures 1,2 in Trefethen's book are a great way of reviewing the material, and I followed them in class. Review carefully any concepts you are not completely familiar with, by going back to your linear algebra textbook if needed. A good reference for basic linear algebra is Halmos' book "Finite dimensional vector spaces". The connection between matrices, linear operators, bases, changes of bases are sometimes not emphasized enough in linear algebra courses, but are fundamental for this course. Go through examples and short exercises proposed in class and do them.
Study and practice Matlab! I recommend watching the tutorials on the mathwork website or read other tutorials (there are many links on the Mathwork website, as well as from other sites). The best way to test what you have learned and consolidate is to look at existing code and code something yourself. You may look at the code I used in the first lecture, and make sure you understand all the steps. Do the

following: if you are not sure how to do them or not sure if you have done them correctly you probably have not read the Matlab tutorials well enough, and/or you have never programmed before: come to my office to discuss:

- Write a Matlab function `function A = Fibonacci( N )` that takes as input N, and returns in A the first N terms of the Fibonacci sequence. Test your program. Write a script that asks for the input N and prints out the first N Fibonacci numbers by calling the `Fibonacci` function.
- Write a Matlab function `function A = HilbertMatrix( m )` that constructs the $m \times m$ matrix A with $A_{ij} = \frac{1}{i+j-1}$. Test your program. Then write a script that starts with a random m vector, and applies the matrix A (obtained by calling `HilbertMatrix(m)`) to it N times, saving the result of the n-th multiplication as the n column of a matrix V. Do this for, say, $N = 10$ or $N = 20$. What can you say about the columns of V? How does their length vary with n? What is happening to the angles between the columns? The last column is $V(:, N)$: construct the unit length vector $v = \frac{V(:,N)}{||V(:,N)||}$, and look at the angle between Av and v: is it small, large, neither? Based on this, it looks like v is pretty close to being an ... of A. Does the result depend on the initial random vector? Check.

Exercises

Exercise 1 (Inspired by exercises in Lecture 1 in Trefethen's book). Let A be a 4×4 matrix to which we apply the following operations:

1. multiply row 3 by 2,
2. add column 1 to column 3,
3. replace column 2 with column 1
4. swap row 1 and row 3
5. delete column 2, reducing the column dimension to 3.

Write the result as a product of 5 matrices. Then write it again as the product of 3 matrices BAC (A as above). Could the result be written only as a product BA or a product AC?

Exercise 2 (From Lecture 1 in Trefethen's book). We say that a matrix R is upper-triangular if $R_{ij} = 0$ for $i > j$. Show that the inverse R^{-1} is also upper-triangular.