Homework Sheet 2

VU Numerical Algorithms, WiSe 2021

due date: 18.11.2021, 18:00

Basic Guidelines - please read carefully!

- Your homework report is *very important* for the grading of your homework. Your report has to provide a clear, well structured, compact and understandable summary of what you did for your homework and what the results are (including *well explained* and *understandable* figures of your experimental results).
- For the programming parts, a correct implementation is *not* sufficient. If you do not provide a clear and well readable summary of your implementation and of your experiments, you will not get a good grade on the programming part.
- Due to the number of students in class, we cannot debug your code in case it produces wrong results. As a consequence, incorrect code in a programming part will always lead to zero points for this part, since it is not possible to find your programming errors and distinguish between different implementation errors.
- In total, your report for this homework sheet may at most have *five pages*. Anything beyond the fifth page will not be considered in the grading.

Paper-and-Pencil Exercises

1. (*2 points*) We have discussed elementary elimination matrices M_k in class. Prove the following two properties of elementary elimination matrices, which are very important for making LU factorization work efficiently in practice:

Property (a): M_k is nonsingular.
Hint: Represent M_k^{-1} explicitly and then show that $M_k M_k^{-1} = M_k^{-1} M_k = I$.

Property (b): The product of two elementary elimination matrices M_k and M_j with $k \leq j$ is essentially their "union"; and therefore they can be multiplied without any computational cost.

2. (*2 points*) We know that a matrix A is singular if $\det(A) = 0$. Can we also conclude that the determinant of a matrix is a good indicator of near singularity? In other words, does the magnitude of a nonzero determinant give any information about how close to singular the matrix is?

   Give a proof for your answer.

Hints: Experimental investigations may help for getting a first idea. It may also help to consider simple cases, such as multiples of the identity matrix.

Programming Exercises

Note

Please stick carefully to the interface described below. This is necessary for us to evaluate your code, which also means that you will not get points for any piece of code that fails to run for interface related reasons.

If you have not already done so, *read the homework report guidelines* on moodle. The aspects described therein form the basis for grading the formal aspects of the report.

Prerequisites

1. Basics:
   - Please use Octave[1] *version 4.4* or higher and indicate the Octave version in your report. Your submission will be evaluated.
   - Do not import additional packages and do not use global variables.
   - Pay attention to the interface definitions, i.e., use the specified terms. In/output parameters must be in the specified order.
   - Do not exploit any special structure in the input data. Your routines must be generic and have to work for all $n > 1$.
   - Do not use any existing code which you did not write yourself! However, for solving linear systems in this assignment, feel free to use Octave routines.
   - You can define your own routines in order to write modular code but please stay consistent with the predefined interface.
   - Further, you can add more parameters, so you do not need to recompute parts of the program and accelerate your computation. However, you should add default parameters, so the program still works with the predefined parameters given in the assignment description.
2. Interface:
   - Mandatory for **all Parts**:
     a) Write a script `assignment2.m` to test your routines and plot your results.[2]
     b) Use Octave's `rand` function wherever data is randomly generated and set no seed.
   - Mandatory for **Part I**:
     a) Create a file `magnitude.m` of the following form:

```
m = magnitude(x, y)
```

[1] Octave download page: `https://www.gnu.org/software/octave/download.html`
[2] In case you do not use Octave for plotting, use `assignment2.m` to export the data plotted in `.csv` files.

- Input: x and y are scalars
- Output: scalar m, with

$$m := \frac{\max(x, y)}{\min(x, y)}.$$

Remark: Use this routine to assess the accuracy of your condition estimator.

b) Create a file `est_cond.m` for the condition number estimator:

```
c = est_cond(A)
```

- Input: $n \times n$ matrix A
- Output: scalar c which is the estimated 1-norm condition number of A

- Mandatory for **Part II**:

a) Create a file `rhs_perturbation.m` of the following form:

```
[E, delta_b] = rhs_perturbation(n)
```

- Input: The dimension n
- Output: Random $n \times n$ matrix E , random $n \times 1$ vector *delta_b*, both of 1-norm 10^{-8}.

b) Create a file `lhs_perturbation.m` of the following form:

```
[x, delta_x] = lhs_perturbation(A, E, b, delta_b)
```

- Input: $n \times n$ matrices A and E, $n \times 1$ vectors b and *delta_b*
- Output: $n \times 1$ vector x which is the solution to the linear system $Ax = b$, $n \times 1$ vector *delta_x* which is the difference $\hat{x} - x$ between x and the solution $\hat{x}$ to the perturbed linear system.

c) Create a file `bounds.m` of the following form:

```
[lb, rb] = bounds(A, E, b, delta_b)
```

- Input: $n \times n$ matrices A and E, $n \times 1$ vectors b and *delta_b*
- Output: scalar *lb* containing the left hand side, scalar *rb* containing the right hand side of the bound (2) in the assignment text.

3. Submission:

- Upload a single zip archive with all your source code files and your report (as a single PDF file named *report.pdf* with all plots and discussions of results as well as your solution to the paper-and-pencil exercises) on the course page in Moodle.
- Name your archive `a<matriculation number>_<last name>.zip` (e.g. *a01234567_Mustername.zip*)
- Directories in the archive are not allowed
- A complete submission should include the following files *at the top level of the zip file*:

a) Routines: `magnitude.m, est_cond.m, rhs_perturbation.m, lhs_perturbation.m, bounds.m` and self defined routines (optional)

b) Script: `assignment2.m`

c) Documentation: *report.pdf*

Part I – A Condition Estimator (8 points)

Take a look at the condition estimation algorithm implemented in LINPACK and described in [1] and [2].

1. (*3 points*) Explain how and why this condition estimation algorithm works.
2. (*5 points*) Implement this algorithm and compare it experimentally to `cond` and `condest` in terms of runtime and quality of the condition estimation. In particular, for varying matrix dimension up to $n = 500$ (random dense matrices), plot the difference in magnitude

$$\mathrm{magn}(x, y) = \frac{\max(x, y)}{\min(x, y)} \tag{1}$$

   between the estimated condition number x (from both your implementation and `condest`) and the "exact" condition number y (from `cond`).[3] Why is the quantity (1) informative?

Your report has to contain:

1. A paragraph (about 10 lines) which explains the algorithm.
2. A figure containing the comparison of the quality of the two estimators
3. A figure containing the runtime comparison of `cond`, `condest` and your implementation
4. A brief discussion of your results

Part II - Average Case Perturbation Errors (4 points)

We have seen in class that the relative error in the solution of a linear system due to perturbations E in the matrix A and $\triangle b$ in the right hand side b can be bounded as

$$\frac{\|\triangle x\|}{\|x\|} \leq \mathrm{cond}(A) \left(\frac{\|\triangle b\|}{\|b\|} + \frac{\|E\|}{\|A\|} \right). \tag{2}$$

This is a *worst-case* bound.

In this exercise your task is to empirically evaluate how tight this analytical bound is and how the relative error in practice relates to this bound. For this purpose, proceed as follows:

- Consider the 1-norm and $n = 100 : 50 : 1500$
- For each n, generate
  - a single random $\triangle b$ with $\|\triangle b\|_1 = 10^{-8}$
  - a single random E with $\|E\|_1 = 10^{-8}$
  - many random A and b
- For each n, compute the averages of the left and right hand sides of the bound (2) over the randomly generated input data A and b.
- Plot the averages of the left and right hand sides of the bound (2) over n.

[3] There is a common pitfall in comparing e.g. the output of `condest` with the output of `cond` - have a look at the documentation of both functions to avoid it.

What are your conclusions?

<u>Your report has to contain:</u>

1. A figure showing the averages of left and right hand sides of the bound (2)
2. A brief discussion thereof

References

[1] Jack J Dongarra et al. *LINPACK users' guide*. SIAM, 1979, pp. 1.11–1.13.

[2] A. K. Cline et al. "An Estimate for the Condition Number of a Matrix". In: *SIAM Journal on Numerical Analysis* 16.2 (1979), pp. 368–375. URL: `http://www.jstor.org/stable/2156842`.