March 2023

Assignment 2: Linear Equation systems

A. Hand calculations

In the A-section of the assignments you solve the problems to solve by hand (paper/tablet and pen). You can use software as a pocket calculator on the side if you like, but you submit the hand calculations.
Note, the methods in this course are not really hand calculation methods, the main objective is rather to understand how the algorithms work (and not primarily to get a solution). Therefore, it's important that you follow the algorithms in your solutions.

1. Given the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 10 \\ 3 & 10 & 16 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 7 \\ 13 \end{pmatrix}$$

   a. Perform LU-decomposition with pivoting to find L, U and P.
   You don't need to explicitly form all the intermediate L_i-matrices in the lecture, but rather follow the "LU-factorization in practice" procedure in the slides.
   b. Solve the equation system $Ax = b$ using forward and backward substitution.
   c. Find (if possible) the Cholesky-decomposition. Follow the algorithm for the Cholesky decomposition. Given the result, what is the conclusion?

2. In previous assignment you worked with a system derived from an electric circuit. Let's say that the input data, the voltages in the right-hand-side, are measured values and the instrument used to measure has 1% accuracy. What accuracy in the solution can we guarantee?
   (You can use Python to compute the condition number of the matrix).

B. Software calculations

3. Download the data file **`network.csv`** from Studium. The file contains a connectivity matrix of size 4720×4720, describing a network. It's generated very much according to the same principles as the little network in assignment 1, but here it is a water-pipe network (and it is much bigger). It's constructed in a way so that equations number 1, 4600 and 4666 have water pressures connected to them (you can see them as water towers connected to these nodes in the network).
   If we solve the $Ax = b$ the solution x will contain the water pressure in each of the 4720 nodes. The input vector b represents the "input pressure", i.e. zero in all nodes except entries 1, 4600 and 4666.
   Your task here is to investigate how the pressures changes for different input-pressures, i.e. solve the equation system repeatedly with different right-hand sides b_i, for $i = 1, \cdots 20$ (twenty equation systems). Here, we are not interested in the solution as such, but rather how the computation time can be improved as a result of using different tools in applied linear algebra.

Solve the equation systems in four different ways (a, b ,c and d below), and time (CPU-time) each solution method.

a. Solve the system $Ax_i = b_i, i = 1, \ldots, 20$ in a loop, using **`linalg.solve`**. For simplicity, let the input pressures in all three nodes be equal to i, i.e. the loop number. Remember that the index of the first element in Python is 0 ($b(1)$ is stored in **`b[0]`** etc.).

b. Repeat the same calculations, but use and *take advantage of*, the LU-decomposition, forward and backward substitution.

c. Repeat the same calculations as in a), but this time without a loop. Store all b_i in a 4720×20-matrix B, where $B = (b_1 \quad b_2 \quad \cdots \quad b_{20})$. Use **`solve`** as in a) but solve the systems $AX = B$ without a loop.

d. Finally, repeat c) but use the so called sparse format and sparse solvers in the package **`scipy.sparse`**. This means that you store the matrices A and B in sparse format and use the sparse equation solver **`spsolve`**.

4. How do we know the matrix is sparse? It's hard to see when we have large matrices. One handy tool is to "spy" on the matrix, to plot only the non-zero elements as e.g. dots. Use **`matlabplotlib.pyplot.spy`** to "spy" on the matrix A (it seems to be a good idea to use the option **`markersize=1`**, otherwise it's hard to see the pattern). Is the matrix sparse? Save the plot and submit it.

5. What is sparse format? How is a matrix stored when we use this format? Create a little sparse matrix, store in the sparse format and print it on the screen. How is it actually stored?