Homework 2

Due date: November 23th

General remark: all graphs should be drawn with markers only (no lines)

1. Implement your own Gauss elimination method with pivoting function to solve a system of linear equations. When possible, try to avoid using loops. Avoid using the following MATLAB commands: '\', inv, triu, tril.
   The function should be saved in a separate file and in the following format:
   sol_direct=guass_elim(A,b,pivot_flag)
   Note that the function should get any size of A and b. It should verify A and b are compatible in size and throw an error if they are not. *pivot_flag* is binary; 1 means 'do pivot'.
   Test your function, load matrix A from *BadGaussMatr.m* from the course site. (*load('BadGaussMatr')*). The exact solution is a vector of ones.
   1.1. Compare the accuracy of the computation results with and without pivoting relative to the exact solution.
   1.2. Compare the computation time of the function with and without pivoting relative to the command A\b using the commands *tic* and *toc*.

2. Implement a function for the computation of the determinant and the inverse of a given matrix with the help of the PLU decomposition (*lu* in MATLAB). Avoid using the following MATLAB commands: '\', det.
   The function should be saved in a separate file and in the following format:
   [my_det, my_inv]=my_det_and_inv(input_mat)
   Note that the function should get any size of input matrix. The function must detect if the matrix is square and throw an error if not.
   Test the function on the following matrices: $A=\begin{pmatrix}2 & 1 & 2\\ 1 & 2 & 3\\ 4 & 1 & 2\end{pmatrix}$, $B=\begin{pmatrix}2 & 3 & -1\\ 1 & -2 & 1\\ -1 & -12 & 5\end{pmatrix}$, $C=\begin{pmatrix}2 & 1 & 2\\ 4 & 1 & 2\\ 1 & 2 & 3\end{pmatrix}$
   What are your conclusions?

3. The function *[A,b]=ODEmodel(n)* in the course site produces a matrix *A* and a vector *b* defining a system of linear equations for a given *n*. Generate a system for *n=8*.
   Implement a function to solve a linear system using an iterative method. Extra respect if you avoid using loops.
   The function should be saved in a separate file and in the following format:
   sol_itr=iterative_solver(G,c,x0,k)
   x0 – initial guess
   G – iteration matrix
   c – permutated b vector
   k – desired number of iterations

sol_itr - solution

Test the function with *x=0̲, k=10,* Using Richardson, Jacobi, and GS methods.

What are your conclusions?

4. Use the function *[A,b]=ODEmodel(n)* in the course site to produces a matrix *A* and a vector *b* for *n=100*.

   Create a function that estimates the number of iterations for a given error threshold and the estimates the error for a given number of iterations: [*est_num_iter, est_error*] = *itr_est(x0,G,c,threshold,k).* use the infinity norm. You may call the function from Q3 if needed.

   x0 – initial guess

   G – iteration matrix

   c – permutated b vector

   threshold – desired threshold

   k – desired number of iterations

   4.1. Calculate the *G* matrix and *c* vector for the Gauss Seidel method and estimate the number of iterations, *k*, needed for $\|x - x^{(k)}\| < 10^{-2}$. Starting with *x0=0̲*.

   4.2. Solve the system with the iterative method (Q3) using *k* iterations. Calculate $\|Ax^{(k)} - b\|$ with the infinity norm. Explain the mismatch, if it exists, between the estimated error and the actual error.

   4.3. Using the function from Q1, solve the system with the direct method. Calculate the error $\|Ax - b\|$ with the infinity norm.

   4.4. Explain the reason for mismatch between the direct and iterative solutions x and $x^{(k)}$, Compare the accuracy and computation time for both methods.

5. Create matrix *A* and vector *b* from ODEmodel(20).

   5.1. Find G matrix for the Jacobi method and its spectral radius. You can use the *eig* function.

   5.2. Estimate how many iterations needed to solve the system with the accuracy $\|x - x^{(k)}\| < 10^{-n}, n = 1{:}\,0.1{:}\,7$ using norm inf. Summarize your findings in graph (k versus n)

   5.3. Solve the system using the function in (Q3), with the number of iterations you estimated (repeat 61 times, the length of *n*). Calculate $\|Ax - b\|$ using norm inf for each solution. Draw $\|Ax - b\|$ values you calculated vs n. Discuss the graph.

6. Write a short section of conclusion (i.e. what have learned from this HW)