# Lab Session 9

**MA581 :** Numerical Computations Lab July-November, 2022 S. Bora

---

1. Use the MATLAB function program `[L,U] = genp(A)` to do the following:

   (a) Find the factors $L$ and $U$ of an $LU$ decomposition of $A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$. What is $A - LU$?

   (b) Solve the system of equations $Ax = b$ where $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ by using the computed $LU$ factorization from `genp` and the programs `rowforward` and `colbackward` in Lab 8 for forward and backward substitution in the correct order. What is the difference of your answer with the correct solution in the 2-norm? Is it of the order of unit roundoff? To see this use $\texttt{norm}(\texttt{x}_\texttt{c} - \texttt{x})$ where $x_c$ is the computed solution and $x$ is the exact solution via hand calculation.

   What can you conclude about GENP from the above algorithm? Can you identify the step at which things start to go wrong?

2. Write a function program `[L,U,p] = gepp(A)` to find a unit lower triangular matrix `L`, an upper triangular matrix `U` and a column vector `p` satisfying `A(p,:) = LU` via Gaussian Elimination with Partial Pivoting (GEPP) When doing so please note the following:

   (a) Your code should make only the most minimal changes to the `genp` code written in Lab 1. In particular it should retain all important features of `genp` that ensure efficiency.

   (b) The built in Matlab function program `lu` performs GEPP and GECP to find $LU$ decompositions of appropriately permuted matrices, also giving the permutation matrices used in each case. Type `help lu` for details. Compare the output of your `gepp` code with the corresponding outputs of the `lu` program. *The comparision should be performed for several different randomly generated matrices (use* `randn` *command for this).*

3. Write a function program `x = geppsolve(A,b)` to solve a system $Ax = b$ via GEPP. Your program should call the program `[L,U,p] = gepp(A)` and the programs written in Lab 1 for solving upper and lower triangular systems. Then perform the following experiments.

   (a) Compare your answers with that of the MATLAB command $A\backslash b$ (which uses GEPP to solve the system) for several different choices of $A$ and $b$ that are randomly generated by using the `randn` command. If $x$ and $\widehat{x}$ be the solutions from `geppsolve.m` and $A\backslash b$ respectively, you should use $\texttt{norm}(\texttt{x} - \widehat{\texttt{x}})$ to see the difference. It is expected that $\texttt{norm}(\texttt{x} - \widehat{\texttt{x}}) \approx \texttt{O}(\texttt{u})$.

   (b) Repeat the experiments in question 1 by using `gepp.m` and `geppsolve.m`. This time you should check $\texttt{A}(\texttt{p},:) - \texttt{LU}$ in part (a) and $\texttt{norm}(\texttt{x}_\texttt{c} - \texttt{x})$ in part (b) where $x_c$ is the computed answer via `geppsolve.m` What is your conclusion about the performance of the two Gaussian elimination versions for this $2 \times 2$ system?

4. Given $A \in \mathbb{R}^{n \times n}$, write a function program `d = mydet(A)` that uses an *efficient* version of LU factorization to compute the determinant of $A$ in $O(n^3)$ flops.