Part A Numerical Analysis, Hilary 2018. Problem Sheet 3

1. By performing Gauss Elimination (without pivoting), solve

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \\ 24 \\ 25 \end{bmatrix}.$$

From your calculations, write down an LU factorisation of the matrix A above, and verify that $LU = A$. Then by successive back and forwards substitutions (and without further factorisation) solve $Ax = b_2$ where $b_2 = [4\ 7\ 9\ 2]^T$.

2. What is the determinant of the matrix A in the question above? (Note one of the few algebraic properties of the determinant is that $\det(BC) = \det(B)\det(C)$ and you might also want to consider what is the determinant of a triangular matrix).

3. Given an LU factorisation of a matrix A, how might one calculate a column of the inverse of A? Estimate the computational work in calculating A^{-1} and hence in solving $Ax = b$ via explicit computation of A^{-1} and multiplication by b.

Are you now convinced that this is *not* the way to solve linear systems of equations in practice?!

An even worse technique would be to apply GE separately for each column: what would the computational cost be then?

4. [M] In MATLAB the 'backslash', \, solves linear systems of equations via Gauss Elimination with partial pivoting: thus `x=A\b` will perform a permuted LU factorisation (sometimes called a PLU factorisation: $PA = LU$) of A and solves $Ly = Pb$ via forward substitution and then $Ux = y$ via back substitution, giving back just the solution vector x.

Verify your solutions to the two linear systems in question above.

5. *(Optional)* Prove that the product of two lower triangular matrices is lower triangular and that the inverse of a non-singular lower triangular matrix is lower triangular. Deduce similar results for upper triangular matrices.

6. Suppose A is a real $n \times n$ matrix with $n \geq 2$ and that the permutation matrix

$$P = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & . & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}.$$

Show that premultiplication of A by P reverses the order of the rows of A.

If $A = LU$ is an LU factorisation of A (without pivoting), what is the structure of PLP? Hence describe how to calculate a factorisation $A = \hat{U}\hat{L}$ where $\hat{U}$ is unit upper triangular and $\hat{L}$ is lower triangular.

7. Perform an LU factorisation with partial pivoting on the matrix

$$A = \begin{bmatrix} 2/3 & 1 & 3 \\ 2 & 1 & 4 \\ 1 & 3/2 & 4 \end{bmatrix}$$

and write down the permutation matrix P, the unit lower triangular matrix of multipliers and the upper triangular matrix U such that $PA = LU$ which you should verify.

[**M**] Check your answer with MATLAB:
`[L,U,P]=lu(A)`
will compute P, L and U for you - see how much easier! (but you must do this by hand once in your life and if you've correctly done the above then join the club! else look and see why you've gone wrong).

8. *(Optional)* [**M**] Note that `A = randn(7, 7);` creates a random matrix in MATLAB with entries taken from a normal distribution with mean zero and variance 1. Also, you can time a command using `tic` and `toc` as in `tic; [L, U, P] = lu(A); toc`

Using the above, for random matrices of dimensions 2^k for $k = 5, 6, \ldots, 10$, see by what factor the time to compute an LU factorisation grows as you double the dimension. (I wouldn't try for too much larger dimension else you might be waiting a long time! - if you do want to interrupt, then control and c keys pressed together should stop the current computation and return you to a MATLAB prompt).

You might find a `for` loop useful here. Make sure you do not time the creation of the matrix, i.e., keep the `randn` command outside the `tic toc`.

9. *(Optional)* [**M**] Continuing the above question, using the timing for the matrix of dimension $2^{10} = 1024 \approx 10^3$, estimate how many millions of floating point operations (flops) the computer you are working on can compute in 1 second. Hence estimate the dimension of a matrix for which LU factorisation (or equivalently Gaussian elimination) would take 1 year on your computer (if it had enough memory to store all of the numbers).

10. Specimen Exam question

Usually GE is applied to square matrices. Suppose, however, that we apply GE (without pivoting) to a matrix with m rows and n columns with $m > n$. Assume that no zeros on the diagonal are encountered.

(a) Describe the elimination process algorithmically in the style of a computer program, explaining exactly what loops and arithmetic operations are involved in the elimination process.

(b) Describe the matrix factorization that has been accomplished by GE. How have we decomposed A? Be sure to be explicit about the dimensions of matrices involved, zeros vs. nonzero entries, etc.

(c) Let b be a column vector of dimension m. Depending on b, the system of equations $Ax = b$ may or may not have a solution. Using the results of the Gaussian elimination, show how to compute a $n-$vector x that is the solution, if it exists. Give a condition on x in terms of the results of Gaussian elimination that determines whether or not $Ax = b$ has a solution.

(d) Suppose now that a zero on the diagonal is encountered after all in the elimination process, at step k. Show that the upper left $k \times k$ block of A is singular.