Exercise Sheet 3

1. How can a general matrix factorization $A = BC$ help you to compute $\det(A)$? Consider the following special cases:

   (a) LU factorization: $A = LU$, where $L \in \mathbb{R}^{n\times n}$ is lower triangular with $\ell_{jj} = 1$ for all j, and $U \in \mathbb{R}^{n\times n}$ is upper triangular.

   (b) Cholesky factorization: $A = R^\top R$, where $R \in \mathbb{R}^{n\times n}$ is upper triangular.

   (c) QR factorization: $A = QR$, where $Q \in \mathbb{R}^{n\times n}$ is orthogonal and $R \in \mathbb{R}^{n\times n}$ upper triangular.

2. Show that

   (a) the product of two upper triangular matrices is again upper triangular, and

   (b) the inverse of an upper triangular matrix is again upper triangular.

3. Solve the following linear systems using Gaussian elimination.

$$\text{(a)}\quad \begin{aligned} 2x_1 - 2x_2 + \ \ x_3 &= 6 \\ x_2 + 2x_3 &= 3 \\ 5x_1 + 3x_2 + \ \ x_3 &= 4 \end{aligned} \qquad \text{(b)}\quad \begin{aligned} x_1 - 2x_2 - 3x_3 &= 10 \\ 5x_1 + 6x_2 - \ \ x_3 &= 2 \\ x_1 - \ \ x_2 - \ \ x_3 &= 6 \end{aligned}$$

4. Compute the inverses of the following matrices using Gaussian elimination.

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & -1 \\ 1 & 2 & 5 \end{bmatrix}, \quad \begin{bmatrix} 3 & 1 & -1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

5. Show that the matrix

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

   is positive definite.

6. Check if the following matrices have LU factorizations. If yes, use Gaussian elimination to compute them.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \\ 6 & 0 & 6 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}, \quad \begin{bmatrix} 6 & 2 & 1 \\ 2 & 4 & 1 \\ 1 & 1 & 4 \end{bmatrix}.$$

7. For a regular lower triangular matrix $L \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$, *forward substitution* is the process which solves $Lx = b$ in n steps according to

$$x_j = \Big(b_j - \sum_{k=1}^{j-1} \ell_{jk} x_k\Big)/\ell_{jj}, \quad \text{for } j = 1, \dots, n. \tag{1}$$

   (a) Write a Matlab (or Octave) function that, for given matrix L and vector b, solves $Lx = b$ using forward substitution. Implement the sum in (1) as an inner product.

   (b) Use the built-in `error` function to check if the dimensions of L and b match and if L is regular and lower triangular.

8. For a regular upper triangular matrix $U \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$, *back substitution* is the process which solves $Ux = b$ in n steps according to

$$x_j = \Big(b_j - \sum_{k=j+1}^{n} u_{jk} x_k\Big)/u_{jj}, \quad \text{for } j = n, \dots, 1. \tag{2}$$

   (a) Write a Matlab (or Octave) function that, for given matrix U and vector b, solves $Ux = b$ using back substitution. Implement the sum in (2) as an inner product.

   (b) Use the built-in `error` function to check if the dimensions of U and b match and if U is regular and upper triangular.