# PH307: Introduction to Numerical Analysis

## Tutorial Sheet 9

**Note**: This tutorial sheet deals with numerical methods for solving a system of equations. You have to get the third problem graded from your TA. In addition, there are two more optional programming assignments which you may want to do just to challenge yourself.

1. By means of hand calculations, solve the equation system

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix},$$

   for

   (a) $k_1 = 1, \quad k_2 = -3, \quad k_3 = 5$

   (b) $k_1 = 0, \quad k_2 = 2, \quad k_3 = -2$

   using both the (i) Gaussian Elimination method with pivoting, and (i) Gauss-Jordan elimination method with pivoting. (Gauss-Jordan algorithm is described below.

2. Consider matrix $A$

$$A = \begin{pmatrix} 3 & -1 & 2 \\ 2 & 1 & 1 \\ 1 & -3 & 0 \end{pmatrix}$$

   Obtain its inverse by hand calculations using the Gauss-Jordan elimination technique described below.

3. **Programming Assignment:** Write a computer program to implement the simple Gaussian Elimination method aimed at solving $n$ system of linear equations expressed as

$$AX = b,$$

   where $A$ is the $n \times n$ coefficient matrix, while $X$ and $b$ are the $n$-dimensional column vectors representing the unknown variables and constants. Assume that elements of

$A$, $X$, and $b$ are all real. The program should read the value of $n$, coefficient matrix $A$, and the constant vector $b$, and output the solution vector $x$. The pseudocode for this algorithm is given in the slides of chapter 8. I have tested the pseudocode by writing a program in Fortran 90, and it works just fine. You can check your code for the problem discussed in the lectures with

$$A = \begin{pmatrix} 3 & -1 & 1 & 2 \\ 6 & -4 & 3 & 5 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix}, \; b = \begin{pmatrix} 8 \\ 13 \\ -19 \\ -34 \end{pmatrix}$$

with the solution

$$X = \begin{pmatrix} 3 \\ 1 \\ -2 \\ 1 \end{pmatrix}$$

4. **Optional Programming Assignment #1:** Write a computer program implementing the Gaussian elimnation algorithm with pivoting, which was discussed in a lecture. The pseudocode for this algorithm can be found in the textbook by Cheney and Kincaid.

5. **Optional Programming Assignment #2: Matrix inversion and computation of determinants by Gauss-Jordan elimination method:**

We define three basic row operations.
1. Multiplication of a row by a scalar c. This is achieved by multiplying the matrix with $E_i(c)$, which is defined as a matrix with the i-th diagonal element equal to c, -otherwise the matrix is an unit matrix. For example,

$$E_2(c) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This operation has the effect of multiplying the determinant of a matrix by c.

2. Interchanging i-th and j-th rows. This is accomplished by a matrix $E_{ij}$ which is an otherwise unit matrix but has its i-th and j-th rows interchanged. This multiplies the determinant of the matrix by -1. The matrix which interchanges the 1st and the 4th rows, for instance, is given by

$$E_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

3. Replace the i-th row with the i-th row plus c times the j-th row. This is done by the matrix $E_{ij}(c)$, which is an identity matrix in which the i-th row is replaced by (i-th row + c times j-th row). For example, $E_{23}(c)$ is given by

$$E_{23}(c) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix}$$

This has no effect on the determinant.

The inverse of a matrix A is defined through
$AA_{-1} = A_{-1}A = I$
Thus if we use a set of row operations $B_1$, $B_2$, ... $B_N$ to reduce a matrix A to I, *i.e.*, if
$B_N B_{N-1} \ldots B_2 \, B_1 A = I$
Then on multiplying $A^{-1}$ from the right we get
$B_N B_{N-1} \ldots B_2 \, B_1 = A^{-1}$
The method consists of the following two steps. To conserve storage, we store all stages of A and B in the same array.

(i) Input and initialization: Define the convergence factor $\epsilon$, dimension of the matrix N and set the determinant $\Delta$ value to 1. Define a N x N matrix B which has 1 at the diagonal positions and zero else where.

(ii) Find the pivot element, i.e. the maximum magnitude element in column k on or below the main diagonal. Compare $|A_{k,k}|$, $|A_{k+1,k}|$, ... $|A_{N,k}|$ to find the largest $|A_{max,k}|$. Interchange row imax with row k of the matrices A and B.

(iii) If the maximum element is less than $\epsilon$ exit as the matrix is singular or near singular.

(iv) Normalization of the (k,k) element to 1 by dividing the k-th row by $A_{k,k}$.

$$\Delta = A_{kk}\Delta \tag{1}$$
$$A_{kj} = A_{kj}/A_{kk} \tag{2}$$
$$B_{kj} = B_{kj}/A_{kk} \tag{3}$$

(v) Zero in the off diagonal elements of the k-th column by replacing all the i-th rows (i $\neq$ k) by a suitable linear combination. This has no effect on the determinant.

$$A_{ij} = A_{ij} - A_{ik}A_{kj} \tag{4}$$

$$B_{ij} = B_{ij} - A_{ik}B_{kj} \tag{5}$$

(vi) Test stage controller. If k < N, set k=k+1 and return to step (ii). Otherwise write the output as in step (vi). (vi) Write output:

$A^{-1}$ =B, $|A| = \Delta$.

Write a modular (with subroutines and functions ) program to find the inverse and the determinant of a matrix of arbitrary dimension, by using the Gauss-Jordan elimination technique. In order to check your program, you can use one of the matrices given in this tutorial sheet as input.