FYS 3150/4150: Possible solution of Project 1.

Maxim Kartamyshev

Comments and critics are welcome and can be sent to maximk@fys.uio.no.

1 Part a: Linear equation systems

LU decomposition is one of the commonly used methods to solve systems of linear equation:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_{n-1} \end{pmatrix}$$

which in compact form reads:

$$\mathbf{A}\vec{x} = \vec{y} \tag{1}$$

The method consists of decomposing the original matrix A into a product of lower-diagonal L and upper-diagonal U matrices:

$$A = LU$$

and then solving the linear system (1) in two steps:

$$L\vec{w} = \vec{y} \qquad U\vec{x} = \vec{w} \tag{2}$$

Due to the form of L and U matrices, the two above equations can be easily solved via back substitution.

In this project we use supplied library routines for LU-decomposition and back substitution to solve the following linear system:

$$\begin{pmatrix} -1 & 1 & -4 \\ 2 & 2 & 0 \\ 3 & 3 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \frac{1}{2} \end{pmatrix}$$

which is satisfied with $\vec{x} = [1.25, -0.75, -0.5]^T$. Numerical solution starts with first reading matrix A and vector $\vec{y}$ from a file, LU-decomposition of the matrix with library routine **ludcmp**, and finally obtaining the solution with help of routine **lubksb**. Routine **solve_part_a** in the attached program listing contains implementation of this algorithm and is tested to produce the correct answer.

2 Part b: Tri-diagonal linear systems

LU-decomposition is a general method and can be used for any non-singular matrix A. There also exist less general methods, applicable only to the matrices of a particular form, but allowing for simple and fast computational algorithms.

In this part of the project we have to solve a linear system governed by the so-called tri-diagonal matrix:

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & \dots & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_{n-1} \\ y_{n-1} \end{pmatrix}$$

Instead of using LU-decomposition to solve this problem, a much easier method, accounting for the specific form of matrix A, can be employed:

```
// solve tri-diagonal system
// forward substitution
btemp = b[0]; y[0] = f[0] / btemp;
for(i = 1; i < n; i++) {
  temp[i] = c[i-1] / btemp;
  btemp = b[i] - a[i] * temp[i];
  y[i] = (f[i] - a[i] * y[i-1]) / btemp;
}
// back substitution
for(i = n-2; i >= 0; i--) {
  y[i] = y[i] - temp[i+1] * y[i+1];
}
```

The routine **solve_part_b** in the attached program listing contains implementation of the above algorithm. Solution of the 100×100 tri-diagonal system with $a_i = c_i = -0.01$, $b_i = 0.02$ and $f_i = 0.1*(i+1)$, $i = \overline{0, n-1}$, is displayed on Figure **??**. LU-decomposition method has been used to verify the tri-diagonal solver.

Amount of operations required to solve the $n \times n$ tri-diagonal systems goes as $O(n), n \gg 1$. This stems from the fact that implementation of both forward and back substitution in the algorithm require each only one (non-nested) loop with index i changing from 0 to $n-1$. Compared with $O(n^3)$ operations in LU-decomposition and Gauss elimination, the algorithm for solving tri-diagonal system provides obvious computational advantage, but its applicability is limited to the tri-diagonal matrices only.

It is worth noticing that once one has performed the LU-decomposition of a general non-singular matrix, obtaining solution of corresponding linear system requires only $O(n)$ operations (back-substitution). It means that we can use the LU-decomposed form of the original matrix to solve the whole class of linear system having the same matrix A and different vectors $\vec{y}_i$.

Finally, let's compare times used by the tri-diagonal solver and LU-decomposition method. UNIX function 'time' is used for precise measurements of time intervals. The Table 1 below summarizes the timing results obtained while solving tri-diagonal systems of different dimensionalities n.
The timing results for the tri-diagonal solver show almost no dependence on the dimensionality of the problem. This simply means that the time used by tri-diagonal solver is rather small compared to the time spent in other parts of the program. One can easily see that the LU-decomposition method uses noticeably longer time to solve exactly the same linear system.

Table 1: Time usage, s

n	Tri-diagonal solver	LU-decomposition
1000	4^{-3}	$32 * 10^{-3}$
2000	4^{-3}	$114 * 10^{-3}$
4000	5^{-3}	$442 * 10^{-3}$