First Name: Verena
Last Name: Horak
Date: 08.11.04
Homework Number: 7
Homework Title: Labs Exercise 2

Exercise 2: *Find the solution of a 2-D Laplace Equation, on unit square, with boundary conditions equal to 1 on left and right boundaries and 0 on upper and lower boundaries. Use* $h = 0.2$ *in both dimensions. Use any system solver avalaible. Plot the solution.*

Solution:

We will describe the solution process as it can be implemented using Matlab. This implementation can be found in the Labs' file from lessons in June; running this script will also yield numerical results and confirm conclusions which will be stated here.
We will number the grid points in the usual linear order, starting from the upper left corner and proceeding to the lower right corner.
Laplace-equation is given by $u_{xx} + u_{yy} = 0$, where $u(x, y)$ is the two dimensional solution function to be found.
We will approximate the solution $u(x, y)$ at all inner 16 grid points by replacing the differential equation with a finite difference equation, namely:

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} \;=\; 0 \; .$$

In these equations, $u_{i,j}$ has to be replaced by a boundary value if (i, j) represents an edge point of the grid, which in turn means that either i or j has to be equal to 0 or 5 (for this special case).
In general, note that the grid points (i, j) as well as the approximate values $u_{i,j}$ are numbered in linear order for our Matlab-implementation. Thus, we have to consider values $u_1, u_2, \ldots, u_{16}$, where u_1 corresponds to $u_{1,1}$, u_2 corresponds to $u_{1,2}$ and so on.
Thus, we have a system of 16 linear equations from which to compute the approximate values at all inner grid points. We can rewrite this system in matrix form, which leads us to the following equation of the form $Au = b$,

where the vector b is defined by boundary conditions:

$$\begin{pmatrix}
4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4
\end{pmatrix}
\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \\ u_{15} \\ u_{16} \end{pmatrix}
=
\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

A is a banded matrix that is composed of several submatrices. A quite economic way to define A and b in a Matlab-file may look like this:

```
b = [1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1]';

T = diag([4 4 4 4], 0) + diag([-1 -1 -1], 1) + diag([-1 -1 -1], -1);
I = eye(4);
Z 0 zeros(4, 4);
A = [T -I Z Z; -I T -I Z; Z -I T -I; Z Z -I T];
```

This system of linear equations can be solved using built-in Matlab-functions; for example:

```
u = A\b;
```

The vector u now contains approximate values for all inner grid points in linear order (see above).
In order to plot the solution, we write all values - boundary conditions as well as calculated approximation values - in a single matrix U:

```
U = [ 1, 0, 0, 0, 0, 0, 1;
      1,    u(1:4)',    1;
      1,    u(5:8)',    1;
      1,    u(9:12)',   1;
      1,    u(13:16)',  1;
      1, 0, 0, 0, 0, 0, 1]
surfc(U);  %plot result
```