First Name: Peter
Last Name: Gruber
Date: 08.11.04
Homework Number: 7
Homework Title: Labs Exercise 4

Exercise 4: *Implement 20 steps of Jacobi iteration for the system matrix and boundary conditions from exercise 2. Estimate the maximal error in the iterative solution. 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.
First of all, let's remember what should be done in 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 step size $h = 0.2$ *in both dimensions.*
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 16 grid points by replacing the differential equation with a finite difference equation, namely:

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

From this equation, we find that $u(x, y)$ may be approximated using

$$u_{i,j} = \frac{u_{i-1,j} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j}}{4} \ ,$$

where $u_{i,j}$ may be replaced by boundary values at the outmost points.
The initial approximation is given by boundary values and 0 at all inner grid points:

$$D \ = \ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \ .$$

In Matlab, the matrix D may be defined as follows:

```
D = [ones(1,6);zeros(4,6);ones(1,6)]';
```

The iteration process can be implemented in a simple algorithm, using the approximation formula we've just derived:

```
Dold = D;
for k = 1:20 %implement 20 steps
for i=2:5 %only approximate inner points;
          %boundary values remain untouched
  for j = 2:5
    D(i,j) = (Dold(i-1,j)+Dold(i,j-1)+Dold(i+1,j)+Dold(i,j+1))/4;
  end
end
Dold=D;
end
```

The matrix D now contains the approximate values for all grid points.
A graphical representation can be obtained using the command

```
surfc(D) .
```

From graphics we can see that 20 iterations are sufficient for a nearly exact solution.