Problem Set 5 (10/16, 18, 21)
Due on Fri, Nov 1

1) Let us consider the two-dimensional boundary value problem of a metal plate on $D = \{(x, y); 0 < x, y < 1\}$. The plate temperature $\phi(x, y)$ obeys $-\Delta\phi = 0$ on D with boundary conditions $\phi(x, 1) = 1, \phi(x, 0) = \phi(0, y) = \phi(1, y) = 0$. This means there are no heat sources inside the plate, and one side of the plate is kept at a high temperature while the other three sides are kept at a low temperature.

   We will obtain $\phi(x, y)$ inside the plate. Let $\mathbf{w}$ be the numerical solution, i.e., $w_{ij} \approx \phi(x_i, y_j)$. By the finite-difference scheme, we have $(D_+^x D_-^x + D_+^y D_-^y)w_{ij} = 0$ with mesh size $h = \frac{1}{n+1}$, for $h = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$. This yields a linear system $A\mathbf{w} = \mathbf{f}$. The mesh points are given by $(x_i, y_j) = (ih, jh)$, $i, j = 0, 1, \ldots, n, n + 1$. The finite-difference equations can be written in component form as

$$\frac{1}{h^2}(4w_{ij} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1}) = f_{ij}.$$

   Thus for example by Jacobi's method, we have

$$\frac{1}{h^2}(4w_{ij}^{(k+1)} - w_{i+1,j}^{(k)} - w_{i-1,j}^{(k)} - w_{i,j+1}^{(k)} - w_{i,j-1}^{(k)}) = f_{ij},$$

   where $w_{ij}^{(k)}$ is the numerical solution at step k.

   (a) Solve the problem by Jacobi's method and the Gauss-Seidel method. Do not form the full matrix A (because it is sparse and that would be inefficient). If you like, you can use the MATLAB template below. Submit a copy of the code.

   (b) For each value of h, plot the computed temperature w_{ij} at the final step (including the boundary values) using a contour plot and a mesh plot (type `help contour` and `help mesh` for instructions).

   (c) Present the following results in a table. column 1: h, column 2: number of iterations needed to reach the stopping criterion.

   (d) What is the value of the temperature at the corners of the plate in the limit $h \to 0$? Explain your answer.

   A few tips: (1) Since MATLAB doesn't accept zero indices, take `i=1:n+2, j=1:n+2`. (2) In the case of the two-point boundary value problem in one dimension, we put the boundary values in $\mathbf{f}$. However in a two-dimensional problem, it is more convenient to keep the boundary values in $\mathbf{w}$. Therefore the boundary values and interior values of $\mathbf{w}$ are set at the initial step and the interior values are updated at every new step. The boundary values can be stored in elements of $\mathbf{w}$ with indices `i=1,n+2, j=1,n+2`. (3) The interior values are set to zero at the initial step. (4) You can use the stopping criterion $\|\mathbf{r}_k\|_2/\|\mathbf{r}_0\|_2 \leq 10^{-4}$, where $\mathbf{r}_k = \mathbf{f} - A\mathbf{w}_k$ is the residual at step k.

```
function bvp2d
clear; clf;
tol = ...; % set tolerance for stopping criterion
for icase=1:3
    n = 2^(icase+1)-1; h = 1/(n+1); % set mesh size
    x = 0:h:1; y = 0:h:1; % create x and y arrays for plots
% initialize solution and residual arrays
    w_new = zeros(n+2,n+2);
    w_old = zeros(n+2,n+2);
    res = zeros(n+2,n+2);
% set nonzero boundary values
    for j = ...; w_new(...,...)  = ...; w_old(...,...)  = ...; end
% initialize control variables
    k = 0; ratio = 1;
% start iteration
    while ratio > tol
         k = k+1;
% compute residual vector
         for i = ...; for j = ...;
              res(i,j) = ...;
         end; end
% compute ratio of residual norms
         rn(k) = norm(res,'fro');
         ratio = rn(k)/rn(1);
% compute numerical solution
         for i = ...; for j = ...;
              w_new(i,j) = ...;
         end; end;
         w_old = w_new; % reset numerical solution for next step
    end % end while
% store results for output
    table(icase,1) = h; table(icase,2) = k;
% draw contour plot
    subplot(2,3,icase)
    contour(x,y,w_new); axis square
    string = sprintf('h=1/%d',n+1); title(string)
% draw surface plot
    subplot(2,3,3+icase)
    mesh(x,y,w_new)
    string = sprintf('h=1/%d',n+1); title(string)
end
table
```