```
%
% Poisson equation on a square with Dirichlet boundary conditions
%
function math571_bvp2d
clear; clf;
for p = 2:5
n = 2^p - 1; N = n^2; h = 1/(n+1);
A = zeros(N,N);
for i = 1:N; A(i,i) = 4; end
for i = 2:N; A(i,i-1) = -1; A(i-1,i) = -1; end
for i = n+1:N; A(i,i-n) = -1; A(i-n,i) = -1; end
for i = 2:n; A((i-1)*n+1,(i-1)*n) = 0; A((i-1)*n,(i-1)*n+1) = 0; end
b = zeros(N,1); b(N-n+1:N) = ones(n,1);
x = math571_cholesky_solve(A,b);
%
% For plotting purposes, we transform the solution vector x into a 2D array u.
%
u = zeros(n+2,n+2); u(n+2,1:n+2) = ones(1,n+2);
for i = 1:n; for j = 2:n+1; u(j,i+1) = x(i+(j-2)*n); end; end
if p==2;
    figure(1); subplot(2,2,1); contour( [0:h:1] , [0:h:1] , u ); title('h=1/4')
    figure(2); subplot(2,2,1); mesh( [0:h:1] , [0:h:1] , u ); title('h=1/4')
end
if p==3;
    figure(1); subplot(2,2,2); contour( [0:h:1] , [0:h:1] , u ); title('h=1/8')
    figure(2); subplot(2,2,2); mesh( [0:h:1] , [0:h:1] , u ); title('h=1/8')
end
if p==4;
    figure(1); subplot(2,2,3); contour( [0:h:1] , [0:h:1] , u );
     title('h=1/16')
    figure(2); subplot(2,2,3); mesh( [0:h:1] , [0:h:1] , u ); title('h=1/16')
end
if p==5;
    figure(1); subplot(2,2,4); contour( [0:h:1] , [0:h:1] , u );
     title('h=1/32')
    figure(2); subplot(2,2,4); mesh( [0:h:1] , [0:h:1] , u ); title('h=1/32')
end
end
%
function x = math571_cholesky_solve(A,b)
%
% fill in this function using the algorithm on page 55 of the notes
```