Matlab Program for Second Order FD Solution to Poisson's Equation

Code:

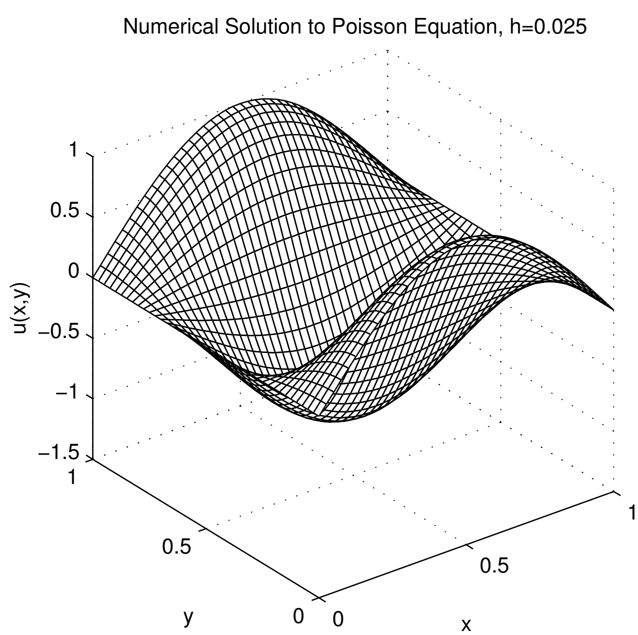
```
% Numerical approximation to Poisson's equation over the square [a,b]x[a,b] with
% Dirichlet boundary conditions.  Uses a uniform mesh with (n+2)x(n+2) total
% points (i.e, n interior grid points).
% Input:
%     ffun : the RHS of poisson equation (i.e. the Laplacian of u).
%     gfun : the boundary function representing the Dirichlet B.C.
%      a,b : the interval defining the square
%        m : m+2 is the number of points in either direction of the mesh.
% Ouput:
%        u : the numerical solution of Poisson equation at the mesh points.
%      x,y : the uniform mesh.
%
function [u,x,y] = fd2poisson(ffun,gfun,a,b,m)

h = (b-a)/(m+1);   % Mesh spacing

[x,y] = meshgrid(a:h:b);   % Uniform mesh, including boundary points.

idx = 2:m+1;
idy = 2:m+1;

% Compute boundary terms, south, north, east, west
ubs = feval(gfun,x(1,1:m+2),y(1,1:m+2));     % Include corners
ubn = feval(gfun,x(m+2,1:m+2),y(m+2,1:m+2)); % Include corners
ube = feval(gfun,x(idy,m+2),y(idy,m+2));     % No corners
ubw = feval(gfun,x(idy,1),y(idy,1));         % No corners

% Evaluate the RHS of Poisson's equation at the interior points.
f = feval(ffun,x(idy,idx),y(idy,idx));

% Adjust f for boundary terms
f(:,1) = f(:,1) - ubw/h^2;             % West
f(:,m) = f(:,m) - ube/h^2;             % East
f(1,1:m) = f(1,1:m) - ubs(idx)/h^2;    % South
f(m,1:m) = f(m,1:m) - ubn(idx)/h^2;    % North

f = reshape(f,m*m,1);

% Create the D2x and D2y matrices

% Full matrix version. Can be made faster with Matlab's sparse library.
z = [-2;1;zeros(m-2,1)];
D2x = 1/h^2*kron(toeplitz(z,z),eye(m));
D2y = 1/h^2*kron(eye(m),toeplitz(z,z));

% Solve the system
u = (D2x + D2y)\f;

% Convert u from a column vector to a matrix to make it easier to work with
% for plotting.
u = reshape(u,m,m);

% Append on to u the boundary values from the Dirichlet condition.
u = [ubs;[ubw,u,ube];ubn];

end
```

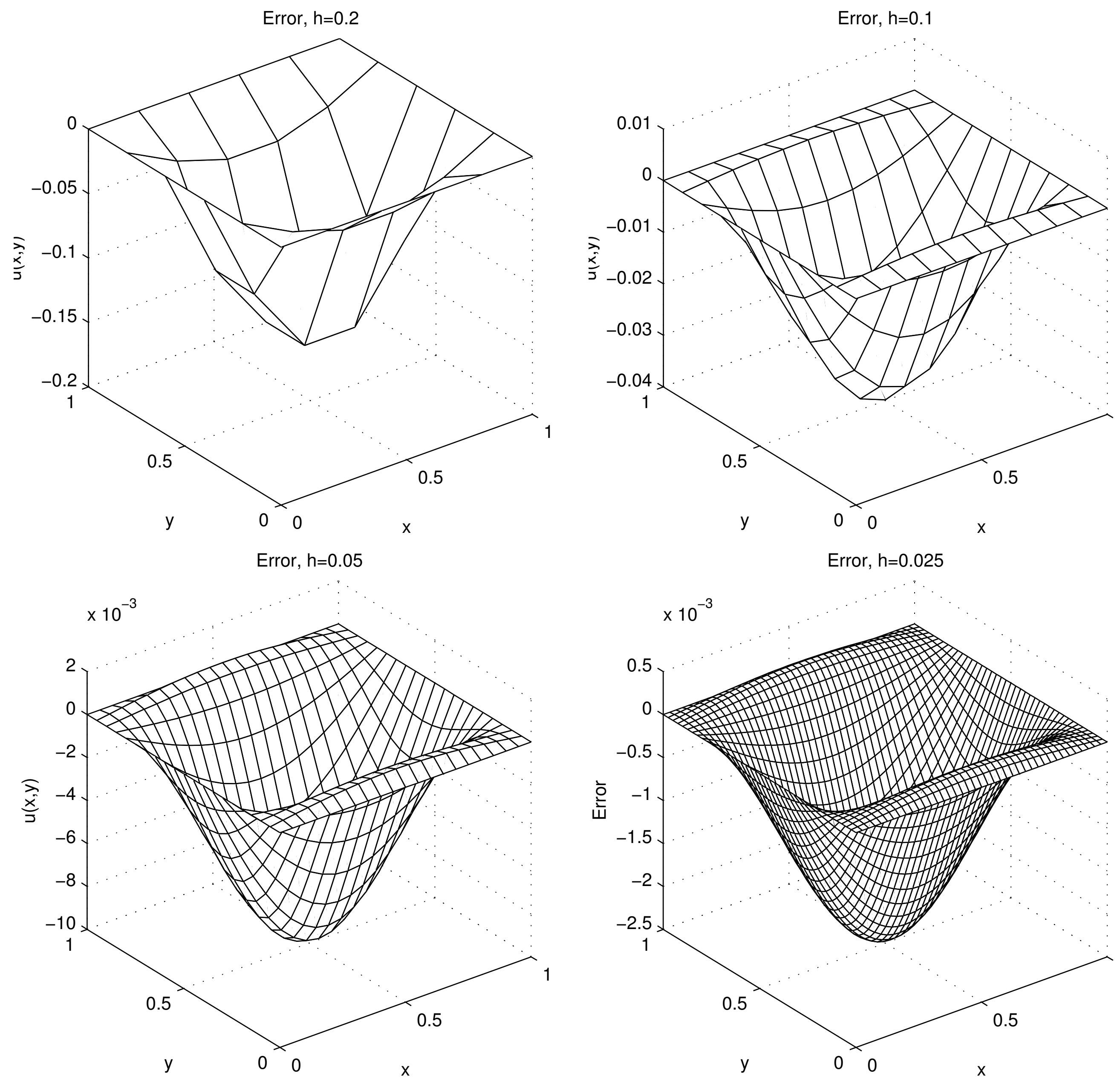
Example:

Use the above MATLAB code to solve the Poisson problem

$$\begin{aligned} \nabla^2 u(x,y) &= -5\pi^2 \sin(\pi x)\cos(2\pi y) \;\text{ for } (x,y) \in \Omega = (0,1)\times(0,1) \\ u(x,y) &= \sin(\pi x)\cos(2\pi y) \;\text{ for } (x,y) \in \partial\Omega \end{aligned}$$

```
% Script for testing fd2poisson over the square [a,b]x[a,b]
a = 0;
b = 1;
k = 6;
m = 2^k-1;  % Number of interior grid points in one direction

f = @(x,y) -5*pi^2*sin(pi*x).*cos(2*pi*y);  % Laplacian(u) = f
g = @(x,y) sin(pi*x).*cos(2*pi*y);          % u = g on Boundary
uexact = @(x,y) g(x,y);                     % Exact solution is g.

% Compute and time the solution
tic
[u,x,y] = fd2poisson(f,g,a,b,m);
gedirect = toc;

fprintf('Direct Gaussian elimination take %d s\n',gedirect);

%% Plot solution
figure, set(gcf,'DefaultAxesFontSize',10,'PaperPosition', [0 0 3.5 3.5]),
surf(x,y,u), xlabel('x'), ylabel('y'), zlabel('u(x,y)'),
title(strcat('Numerical Solution to Poisson Equation, h=',num2str(h)));

% Plot error
figure, set(gcf,'DefaultAxesFontSize',10,'PaperPosition', [0 0 3.5 3.5]),
surf(x,y,u-uexact(x,y)),xlabel('x'),ylabel('y'), zlabel('Error'),
title(strcat('Error, h=',num2str(h)));
```




Error, h=0.2
u(x,y)
0
−0.05
−0.1
−0.15
−0.2
1
0.5
0
y
0
0.5
1
x
Error, h=0.1
u(x,y)
0.01
0
−0.01
−0.02
−0.03
−0.04
1
0.5
0
y
0
0.5
1
x
Error, h=0.05
x 10^−3
u(x,y)
2
0
−2
−4
−6
−8
−10
1
0.5
0
y
0
0.5
1
x
Error, h=0.025
x 10^−3
Error
0.5
0
−0.5
−1
−1.5
−2
−2.5
1
0.5
0
y
0
0.5
1
x