Elliptic Equations - Part II

Model problem we focus on:

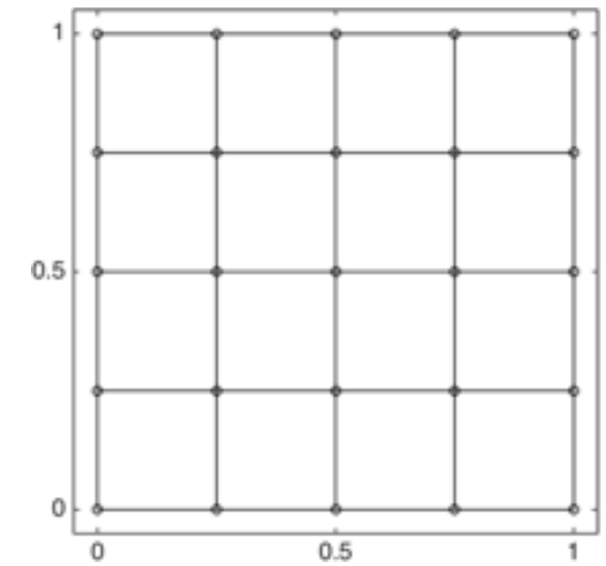
$$\nabla^2 u(x,y) = f(x,y),\ \Omega = \{(x,y) \big| 0 \le x \le 1,\ 0 \le y \le 1\}$$

Boundary conditions $\begin{array}{l} u(x,0) = g_1(x),\ u(1,y) = g_2(y), \\ u(x,1) = g_3(x),\ u(0,y) = g_4(x). \end{array}$

Grid: $(x_j, y_k) = (jh, kh),\ h = 1/(m+1),\ j,k = 0, 1 \ldots, m+1$

$$u_{jk} = u(x_j, y_k)\ f_{jk} = f(x_j, y_k)$$

Example with m=3



Arrange the unknowns and knowns in a matrix:

$$U^h = \begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix} \qquad F^h = \begin{bmatrix} f_{11} & f_{21} & f_{31} \\ f_{12} & f_{22} & f_{32} \\ f_{13} & f_{23} & f_{33} \end{bmatrix}$$

We can then write the discretized equations as

$$\underbrace{\begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix}}_{h^2 D^h_{2,x}} \underbrace{\begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix}}_{U^h} + \underbrace{\begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix}}_{U^h} \underbrace{\begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix}}_{h^2 (D^h_{2,xx})^T} = h^2 \underbrace{\begin{bmatrix} f_{11} & f_{21} & f_{31} \\ f_{12} & f_{22} & f_{32} \\ f_{13} & f_{23} & f_{33} \end{bmatrix}}_{F^h} - \underbrace{\begin{bmatrix} u_{10} + u_{01} & u_{20} & u_{30} + u_{41} \\ u_{02} & 0 & u_{42} \\ u_{03} + u_{14} & u_{24} & u_{34} + u_{43} \end{bmatrix}}_{U^h_{\text{bc}}}$$

This gives the *matrix equation*:

$$D^h_{2,y} U^h + U^h (D^h_{2,x})^T = \underbrace{F^h - \frac{1}{h^2} U^h_{\text{bc}}}_{\widetilde{F}^h}$$

Or the *linear system of equations*:

$$\big(\underbrace{I_m \otimes D^h_{2,y}}_{D_{yy}} + \underbrace{D^h_{2,x} \otimes I_m}_{D_{xx}}\big) \underline{u}^h = \underline{\tilde{f}}^h$$

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix}}_{A^h} \underbrace{\begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \\ u_{21} \\ u_{22} \\ u_{23} \\ u_{31} \\ u_{32} \\ u_{33} \end{bmatrix}}_{\underline{u}^h} = \underbrace{\begin{bmatrix} h^2 f_{11} - u_{01} - u_{10} \\ h^2 f_{12} - u_{02} \\ h^2 f_{13} - u_{03} - u_{14} \\ h^2 f_{21} \qquad - u_{20} \\ h^2 f_{22} \\ h^2 f_{23} \qquad - u_{24} \\ h^2 f_{31} - u_{41} - u_{30} \\ h^2 f_{32} - u_{42} \\ h^2 f_{33} - u_{43} - u_{34} \end{bmatrix}}_{\underline{f}^h}$$

For a general m, A^h is an $m^2 \times m^2$ matrix

Three questions for this method.

1. Is the matrix A^h non-singular for all h?
2. Does the solution converge to the exact solution as $h \longrightarrow 0$?
3. Are there fast ways to solve the system?

Question 1: Is the matrix A^h non-singular for all h?

2. Does the solution converge to the exact solution as $h \longrightarrow 0$?

3. Are there fast ways to solve the system?

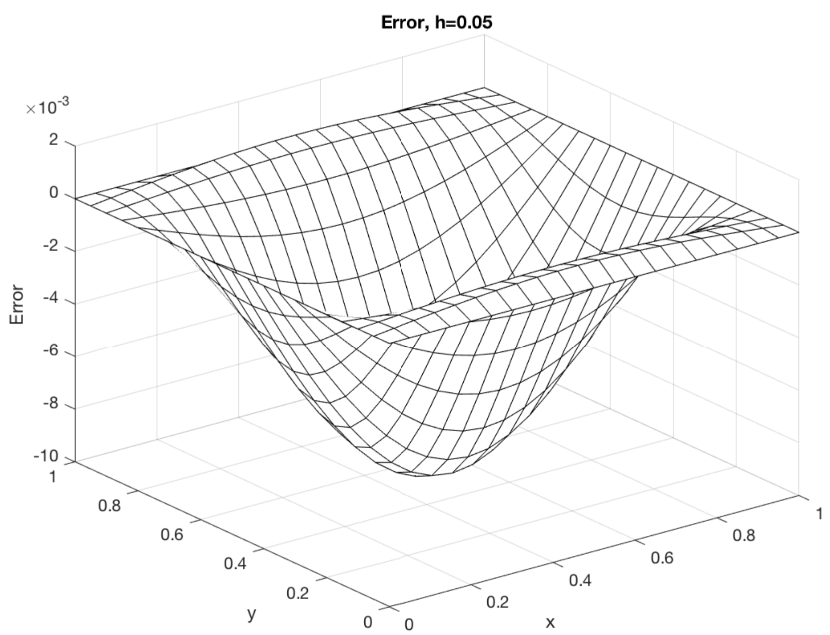
Matlab code that uses Gaussian elimination:

```
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.

% Compute u on the boundary from the Dirichlet boundary condition
ub = zeros(m,m);
idx = 2:m+1;
idy = 2:m+1;
% West and East boundaries need special attention
ub(:,1) = feval(gfun,x(idy,1),y(idy,1));             % West
ub(:,m) = feval(gfun,x(idy,m+2),y(idy,m+2));         % East
% Now the North and South boundaries
ub(1,1:m) = ub(1,1:m) + feval(gfun,x(1,idx),y(1,idx)); % South
ub(m,1:m) = ub(m,1:m) + feval(gfun,x(m+2,idx),y(m+2,idx)); % North
% Convert ub to a vector using column reordering
ub = (1/h^2)*reshape(ub,m*m,1);

% Evaluate the RHS of Poisson's equation at the interior points.
f = feval(ffun,x(idy,idx),y(idy,idx));
% Convert f to a vector using column reordering
f = reshape(f,m*m,1);

% Create the D2x and D2y matrices

% Full matrix version.  This could be made much faster by using Matlab's
% sparse matrix functions (see "spdiags" for more details).
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-ub);

% 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 = [feval(gfun,x(1,1:m+2),y(1,1:m+2));...
    [feval(gfun,x(idy,1),y(idy,1)),u,feval(gfun,x(idy,m+2),y(idy,m+2))];...
     feval(gfun,x(m+2,1:m+2),y(m+2,1:m+2))];
```

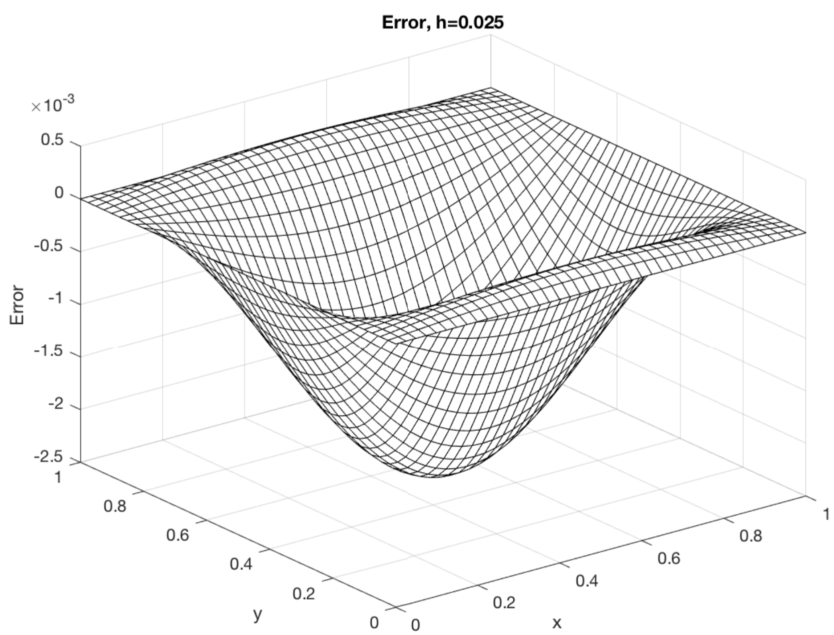
Numerical results

```
% Script for testing fd2poisson over the square [a,b]x[a,b]
a = 0;
b = 1;
m = 19;  % 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.

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

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

% Plot error
figure, set(gcf,'DefaultAxesFontSize',8,'PaperPosition', [0 0 3.5 3.5]),
mesh(x,y,u-uexact(x,y)), colormap([0 0 0]),xlabel('x'),ylabel('y'),
zlabel('Error'), title(strcat('Error, h=',num2str(h)));
```

Plots of the error for $m+1=20$, 40, 80

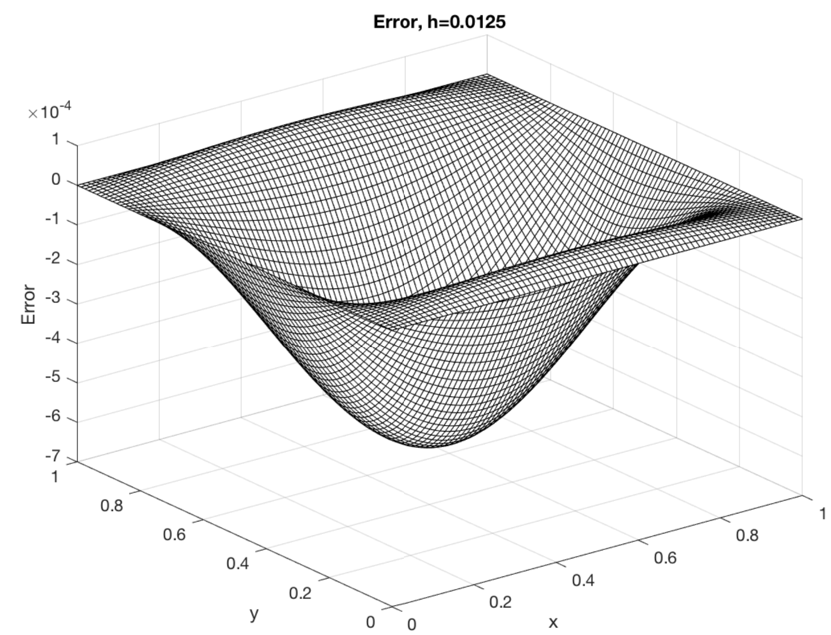


Elapsed time is 0.002523 seconds.



Elapsed time is 0.055546 seconds.

22 times more seconds than $m+1=20$



Elapsed time is 1.744623 seconds.

31 times more seconds than $m+1=40$

Computational cost of Gaussian elimination approach:

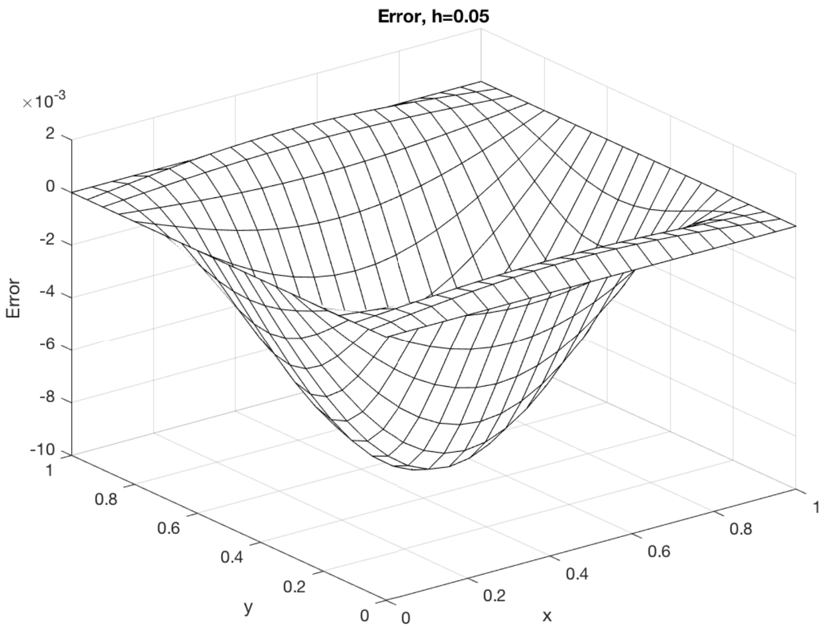
Sparse Gaussian elimination solvers:

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix}}_{A^h} \underbrace{\begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \\ u_{21} \\ u_{22} \\ u_{23} \\ u_{31} \\ u_{32} \\ u_{33} \end{bmatrix}}_{\underline{u}^h} = \underbrace{\begin{bmatrix} h^2 f_{11} - u_{01} - u_{10} \\ h^2 f_{12} - u_{02} \\ h^2 f_{13} - u_{03} - u_{14} \\ h^2 f_{21} \qquad - u_{20} \\ h^2 f_{22} \\ h^2 f_{23} \qquad - u_{24} \\ h^2 f_{31} - u_{41} - u_{30} \\ h^2 f_{32} - u_{42} \\ h^2 f_{33} - u_{43} - u_{34} \end{bmatrix}}_{\underline{f}^h}$$

In general:

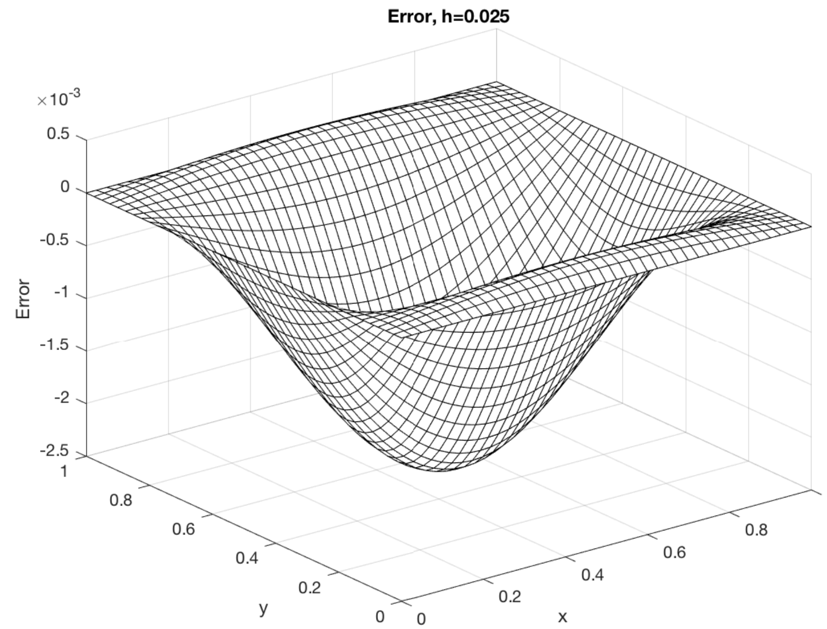
$$A^h = \frac{1}{h^2}\begin{bmatrix} T & I & & & \\ I & T & I & & \\ & I & T & I & \\ & & \ddots & \ddots & \ddots \\ & & & I & T & I \\ & & & & I & T \end{bmatrix}$$

Plots of the error for m+1=20, 40, 80



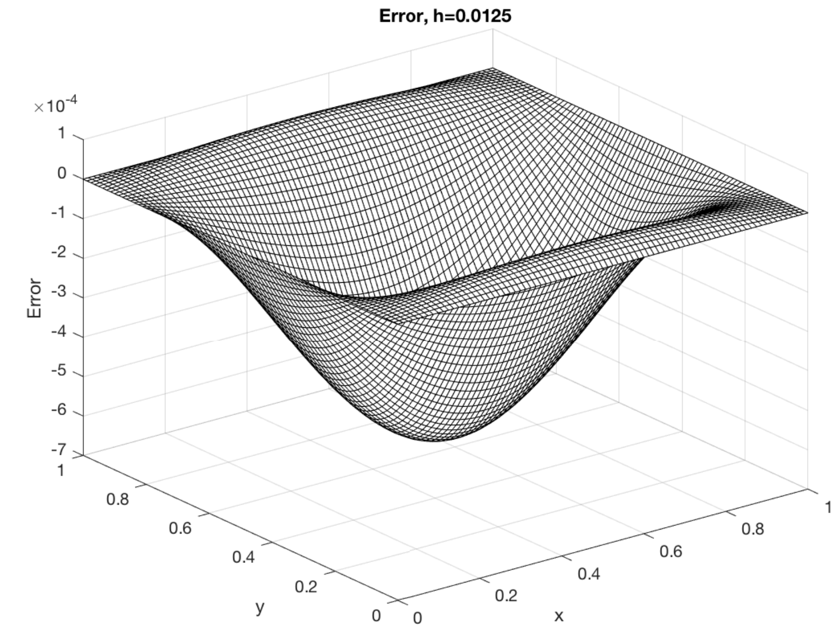
Dense: Elapsed time is 0.002523 seconds.

Sparse: Elapsed time is 0.000719 seconds.



Dense: Elapsed time is 0.055546 seconds.

Sparse: Elapsed time is 0.001734 seconds.



Dense: Elapsed time is 1.744623 seconds.

Sparse: Elapsed time is 0.008137 seconds.

Fast direct solvers based on the FFT:

Recall the matrix equation version of the problem:

$$D^h_{2,y} U^h + U^h (D^h_{2,x})^T = \underbrace{F^h - \frac{1}{h^2} U^h_{\text{bc}}}_{\widetilde{F}^h}$$

Let

$$V^h = \begin{bmatrix} \sin(\frac{\pi}{m+1}1) & \sin(\frac{\pi}{m+1}2) & \cdots & \cdots & \sin(\frac{\pi}{m+1}m) \\ \sin(\frac{\pi}{m+1}2) & \sin(\frac{\pi}{m+1}4) & \cdots & \cdots & \sin(\frac{\pi}{m+1}2m) \\ \sin(\frac{\pi}{m+1}3) & \sin(\frac{\pi}{m+1}6) & \cdots & \cdots & \sin(\frac{\pi}{m+1}3m) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \sin(\frac{\pi}{m+1}m) & \sin(\frac{\pi}{m+1}2m) & \cdots & \cdots & \sin(\frac{\pi}{m+1}m^2) \end{bmatrix} \qquad \Lambda^h = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ & & & & \lambda_m \end{bmatrix}$$

$$\lambda_j = -\frac{4}{h^2} \sin^2\left(\frac{\pi}{2(m+1)} j\right)$$