Contents


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

% Laplacian(u) = f
f = @(x,y) 10*pi^2*(1+cos(4*pi*(x+2*y))-2*sin(2*pi*(x+2*y))).*exp(sin(2*pi*(x+2*y)));
% u = g on Boundary
g = @(x,y) exp(sin(2*pi*(x+2*y)));

% Exact solution is g.
uexact = @(x,y) g(x,y);

% Compute and time the solution
k1    = zeros(1,3);
h1    = zeros(1,3);
m1    = zeros(1,3);
t     = zeros(1,3);
t_sor = zeros(1,3);
t_sp  = zeros(1,3);
t_dst = zeros(1,3);
t_mg  = zeros(1,3);

t1   = [];
tsor = [];
tsp  = [];
tdst = [];
tmg  = [];
for ii = 1:3
    for k=4:6
        k1(k-3) = k;
        m1(k-3) = 2^k-1;
        m = 2^k-1;
        h1(k-3) = (b-a)/(m+1);
        h = (b-a)/(m+1);
        w = 2/(1+sin(pi*h)); %optimal relaxation parameter

        tic
        [u,x,y] = fd2poisson(f,g,a,b,m);
        gedirect = toc;
        t(k-3) = gedirect;

        tic
        [usor,x,y] = fd2poissonsor(f,g,a,b,m,w);
        gedirect = toc;
        t_sor(k-3) = gedirect;

        tic
        [usp,x,y] = fd2poissonsp(f,g,a,b,m);
        gedirect = toc;
        t_sp(k-3) = gedirect;

        tic
        [udst,x,y] = fd2poissondst(f,g,a,b,m);
        gedirect = toc;
        t_dst(k-3) = gedirect;

        tic
        [umg,x,y] = fd2poissonmg(f,g,a,b,m);
        gedirect = toc;
        t_mg(k-3) = gedirect;
    end

    t1   = [t1,t];
    tsor = [tsor,t_sor];
    tsp  = [tsp, t_sp];
    tdst = [tdst, t_dst];
    tmg  = [tmg,t_mg];
end

%k=4
c4=[t1(1);t1(4);t1(7)]';
d4=[tsor(1);tsor(4);tsor(7)]';
e4=[tsp(1);tsp(4);tsp(7)]';
fd4=[tdst(1);tdst(4);tdst(7)]';
h4=[tmg(1);tmg(4);tmg(7)]';

%k=5
c5=[t1(2);t1(5);t1(8)]';
d5=[tsor(2);tsor(5);tsor(8)]';
e5=[tsp(2);tsp(5);tsp(8)]';
fd5=[tdst(2);tdst(5);tdst(8)]';
h5=[tmg(2);tmg(5);tmg(8)]';

%k=6
c6=[t1(3);t1(6);t1(9)]';
d6=[tsor(3);tsor(6);tsor(9)]';
e6=[tsp(3);tsp(6);tsp(9)]';
```

```
fd6=[tdst(3);tdst(6);tdst(9)]';
h6=[tmg(3);tmg(6);tmg(9)]';

k4 = [k1(1);k1(1);k1(1)];
m4 = [m1(1);m1(1);m1(1)];
h4 = [h1(1);h1(1);h1(1)];
%Table showing timing results of each method and for each value of m.
Table4 = table(k4,m4,h4,c4(:),d4(:),e4(:),fd4(:),h4(:), 'VariableNames',{'k','m','h','t_stan','time_sor','time_sp','time_dst','time_mg'});

k5 = [k1(2);k1(2);k1(2)];
m5 = [m1(2);m1(2);m1(2)];
h5 = [h1(2);h1(2);h1(2)];
%Table showing timing results of each method and for each value of m.
Table5 = table(k5,m5,h5,c5(:),d5(:),e5(:),fd5(:),h5(:), 'VariableNames',{'k','m','h','t_stan','time_sor','time_sp','time_dst','time_mg'});

k6 = [k1(3);k1(3);k1(3)];
m6 = [m1(3);m1(3);m1(3)];
h6 = [h1(3);h1(3);h1(3)];
%Table showing timing results of each method and for each value of m.
Table6 = table(k6,m6,h6,c6(:),d6(:),e6(:),fd6(:),h6(:), 'VariableNames',{'k','m','h','t_stan','time_sor','time_sp','time_dst','time_mg'});

Table = [Table4; Table5; Table6]

%mean
Tablem4 = table(k1(1),m1(1),h1(1),mean(c4),mean(d4),mean(e4),mean(fd4),mean(h4), 'VariableNames',{'k','m','h','t_stan','time_sor','time_sp','time
Tablem5 = table(k1(2),m1(2),h1(2),mean(c5),mean(d5),mean(e5),mean(fd5),mean(h5), 'VariableNames',{'k','m','h','t_stan','time_sor','time_sp','time
Tablem6 = table(k1(3),m1(3),h1(3),mean(c6),mean(d6),mean(e6),mean(fd6),mean(h6), 'VariableNames',{'k','m','h','t_stan','time_sor','time_sp','time

Table_mean = [Tablem4; Tablem5; Tablem6]

fprintf(' Make: Ilife Zed AIR plus \n Processor type: Intel Celeron CPU N3350\n Speed: @ 1.10 GHz x2 \n Memory: 6GB DDR III RAM\n');

fprintf(' (d). According to the computed mean wall clock time from Table_mean, fd2poissondst \n appears to be the best since it has the lowest co

fprintf(' Note: I used only k values from 4 to 5, because when i tried to run for k = 7 and above \n the MATLAB on my computer terminated, so i w
```

Table =

9×8 table

k	m	h	t_stan	time_sor	time_sp	time_dst	time_mg
4	15	0.0625	0.3285	0.006445	0.3483	0.13858	0.0625
4	15	0.0625	0.012216	0.002586	0.025811	0.002758	0.0625
4	15	0.0625	0.00337	0.001522	0.00245	0.000893	0.0625
5	31	0.03125	0.094806	0.023273	0.024131	0.008584	0.03125
5	31	0.03125	0.098674	0.007825	0.009372	0.01643	0.03125
5	31	0.03125	0.089086	0.006352	0.006888	0.001161	0.03125
6	63	0.015625	4.9986	0.043212	0.030634	0.016548	0.015625
6	63	0.015625	2.9922	0.038942	0.040301	0.002619	0.015625
6	63	0.015625	4.4111	0.074707	0.056214	0.010564	0.015625

Table_mean =

3×8 table

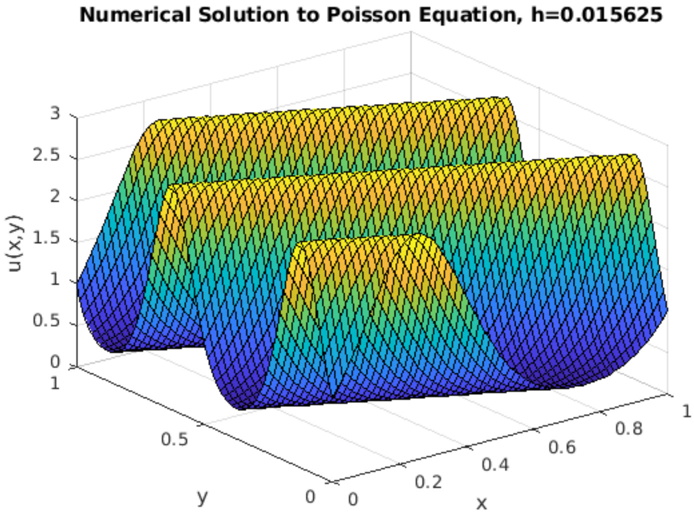
k	m	h	t_stan	time_sor	time_sp	time_dst	time_mg
4	15	0.0625	0.1147	0.0035177	0.12552	0.047412	0.0625
5	31	0.03125	0.094189	0.012483	0.013464	0.008725	0.03125
6	63	0.015625	4.134	0.052287	0.042383	0.0099103	0.015625

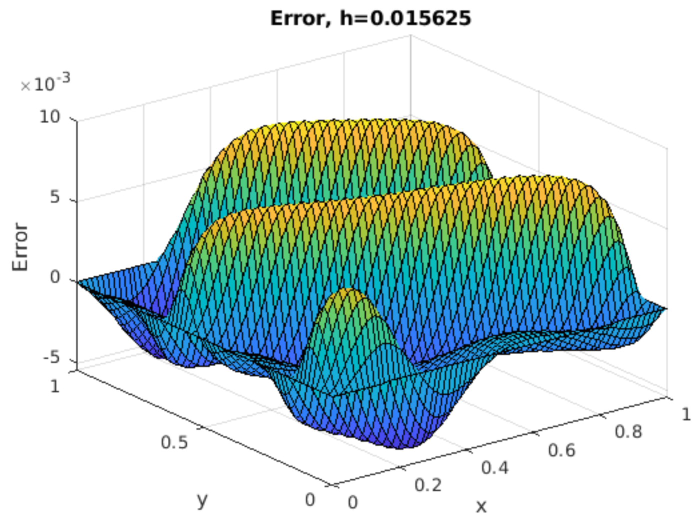
```
 Make: Ilife Zed AIR plus
 Processor type: Intel Celeron CPU N3350
 Speed: @ 1.10 GHz x2
 Memory: 6GB DDR III RAM
 (d). According to the computed mean wall clock time from Table_mean, fd2poissondst
 appears to be the best since it has the lowest computation time  amongest all other method as m increases.
 Note: I used only k values from 4 to 5, because when i tried to run for k = 7 and above
 the MATLAB on my computer terminated, so i wouldnot perform any further simulations beyond k=6.
```

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)));
```

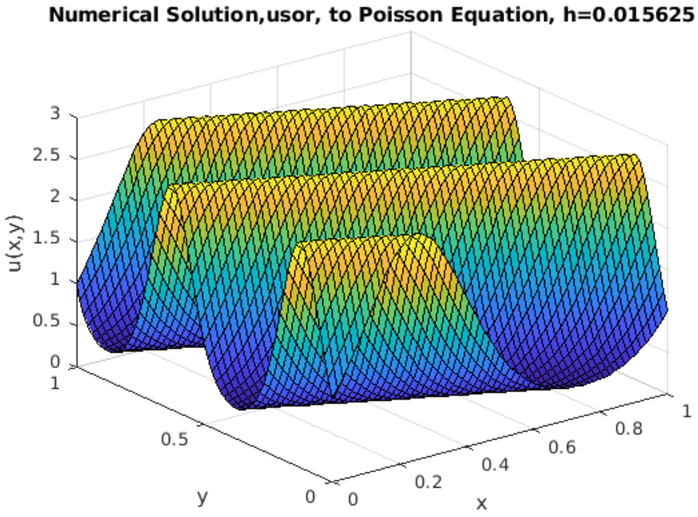


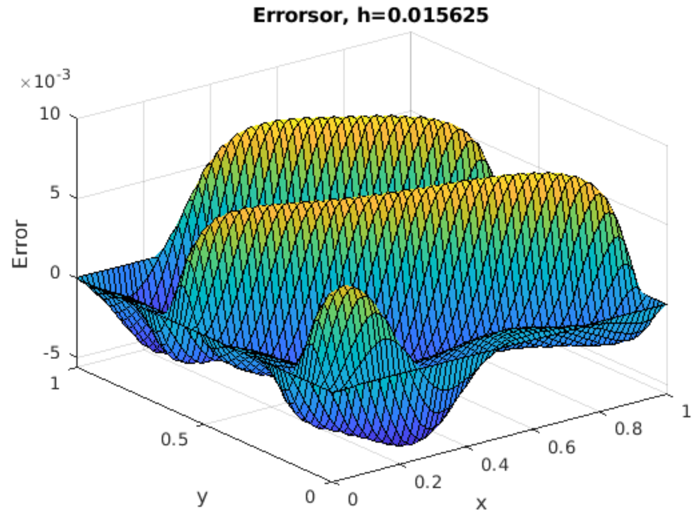


Plot solution

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

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

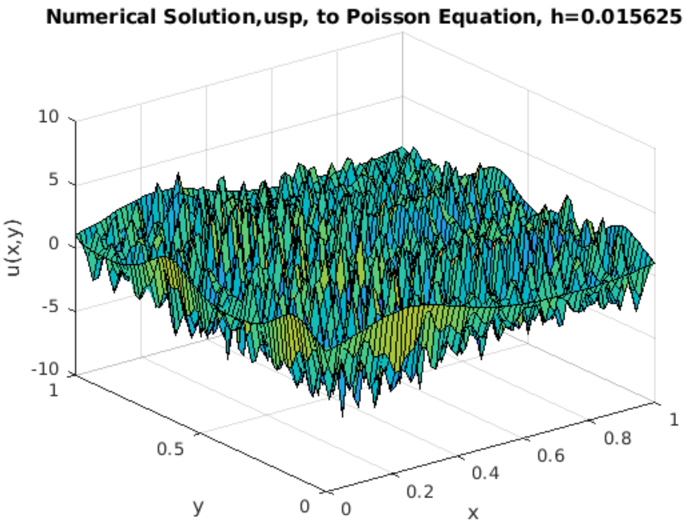


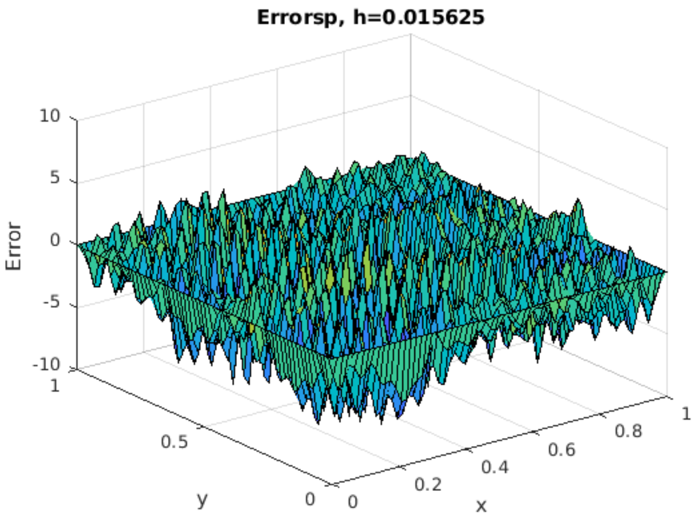


Plot solution

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

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

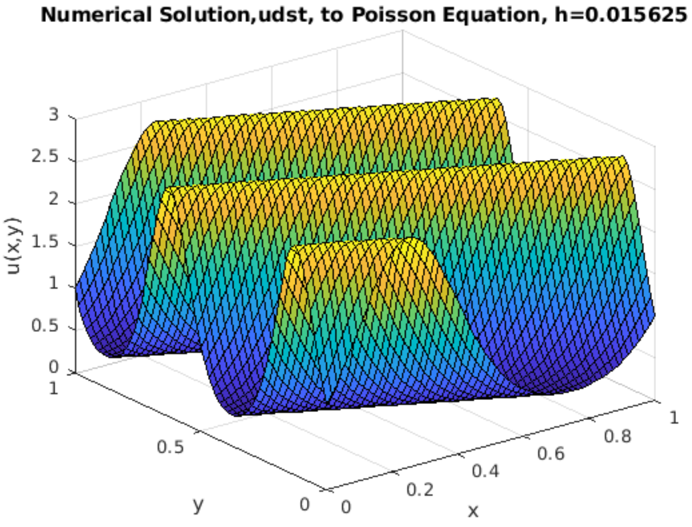


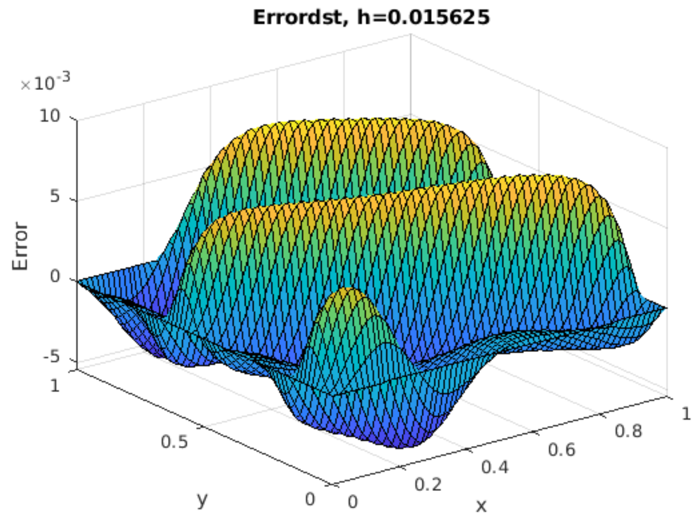


Plot solution

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

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

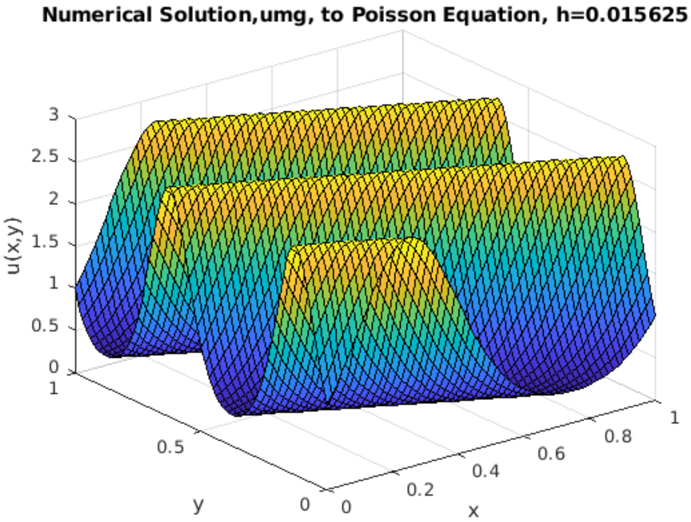


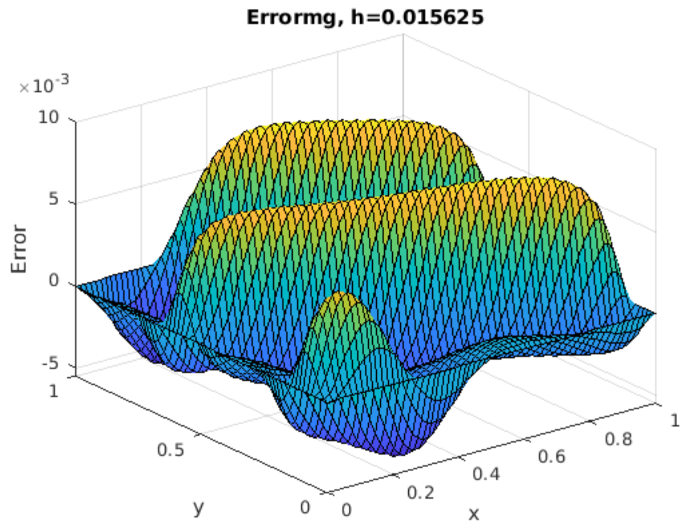


Plot solution

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

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





Published with MATLAB® R2020a