***Problem 1 Programming part***

Use x\_hat = 0.5. Use u(x) = sin(x) for x < x\_hat and u(x) = e^x for x > x\_hat as the exact solution.

Since we have already derived the KKT system, the linear system is hard coded in Matlab.

The solution looks like:



Convergence study on L\_inf norm and L\_2 norm:





**MATLAB CODE**

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% FEM 1D Poisson Interface Problem

% Dirichlet on the left

% Neumann on the right

% Jumps on the interface (0.5)

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

close all;

clear all;

clc

% input parameters

a = exp(0.5) - sin(0.5);

b = exp(0.5) - cos(0.5);

u0 = 0;

u1 = exp(1);

List\_dx = [];

List\_L\_inf\_error = [];

List\_L\_2\_error = [];

for m = [10,20,40,80,160,320,640,1280,2560]

N = 2\*m;

dx = 0.5/(m-1);

List\_dx = [List\_dx;dx];

A = zeros(N-1,N-1);

A\_p = zeros(m-1,m-1);

A\_m = zeros(m-1,m-1);

A\_h = zeros(m,m);

for i = 1:m-1

A\_h(i,i) = A\_h(i,i) + 1;

A\_h(i,i+1) = A\_h(i,i+1) - 1;

A\_h(i+1,i) = A\_h(i+1,i) -1;

A\_h(i+1,i+1) = A\_h(i+1,i+1) + 1;

end

A\_h = A\_h/dx;

A\_p = A\_h(2:m,2:m);

A\_m = A\_h(1:m-1,1:m-1);

B = zeros(1,N-2);

B(m-1) = -1;

B(m) = 1;

A(N-1,1:N-2) = B;

A(1:N-2,N-1) = B';

A(1:m-1,1:m-1) = A\_p;

A(m:N-2,m:N-2) = A\_m;

F = zeros(N-1,1);

F(m-1) = -b/2;

F(m) = -b/2;

F(1) = u0/dx;

F(N-2) = u1/dx;

F(N-1) = a;

syms x;

F(1) = F(1) - (-sin(0)\*dx/2);

for i = 2:m-2

F(i) = F(i) - (-sin(dx\*(i-1))\*dx);

end

F(m-1) = F(m-1) - (-sin(0.5)\*dx/2);

F(m) = F(m) - exp(0.5)\*dx/2;

for i = m+1:N-3

F(i) = F(i) - exp(dx\*(i-1))\*dx;

end

F(N-2) = F(N-2) - exp(1)\*dx/2;

U = A\F;

U\_solution = [u0;U(1:N-2);u1];

x = zeros(N,1);

for i = 1:m

x(i) = (i-1)\*dx;

end

for i = m+1:N

x(i) = (i-2)\*dx;

end

U\_exact = [sin(x(1:m));exp(x(m+1:N))];

L\_inf\_error = max(abs(U\_exact-U\_solution));

L\_2\_error = norm(U\_exact-U\_solution,2);

List\_L\_inf\_error = [List\_L\_inf\_error;L\_inf\_error];

List\_L\_2\_error = [List\_L\_2\_error;L\_2\_error];

end

% plot solution

figure

subplot(1,2,1)

plot(x,U\_solution,'o');

title('numerical solution')

subplot(1,2,2)

plot(x,U\_exact,'o')

title('exact solution')

% Infinity norm convergence plot

figure

p = polyfit(log(List\_dx), log(List\_L\_inf\_error), 1);

plot(log(List\_dx), log(List\_L\_inf\_error), 'o', log(List\_dx), polyval(p,log(List\_dx)));

title(['convergence plot']);

legend('Linf error vs h', [num2str(exp(p(2))) ' \* h\^ ' num2str(p(1))], 'location', 'northwest');

xlabel('log(h)');

ylabel('log(L\_inf\_error)');

% L2 norm convergence plot

figure

p = polyfit(log(List\_dx), log(List\_L\_2\_error), 1);

plot(log(List\_dx), log(List\_L\_2\_error), 'o', log(List\_dx), polyval(p,log(List\_dx)));

title(['convergence plot']);

legend('L2 error vs h', [num2str(exp(p(2))) ' \* h\^ ' num2str(p(1))], 'location', 'northwest');

xlabel('log(h)');

ylabel('log(L\_2\_error)');