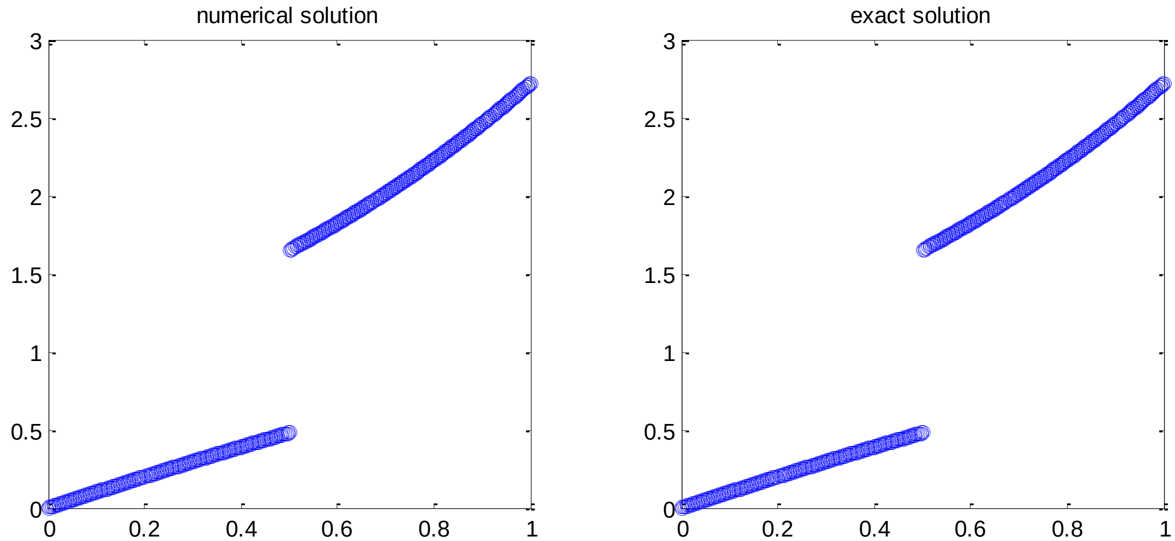
Problem 1 Programming part

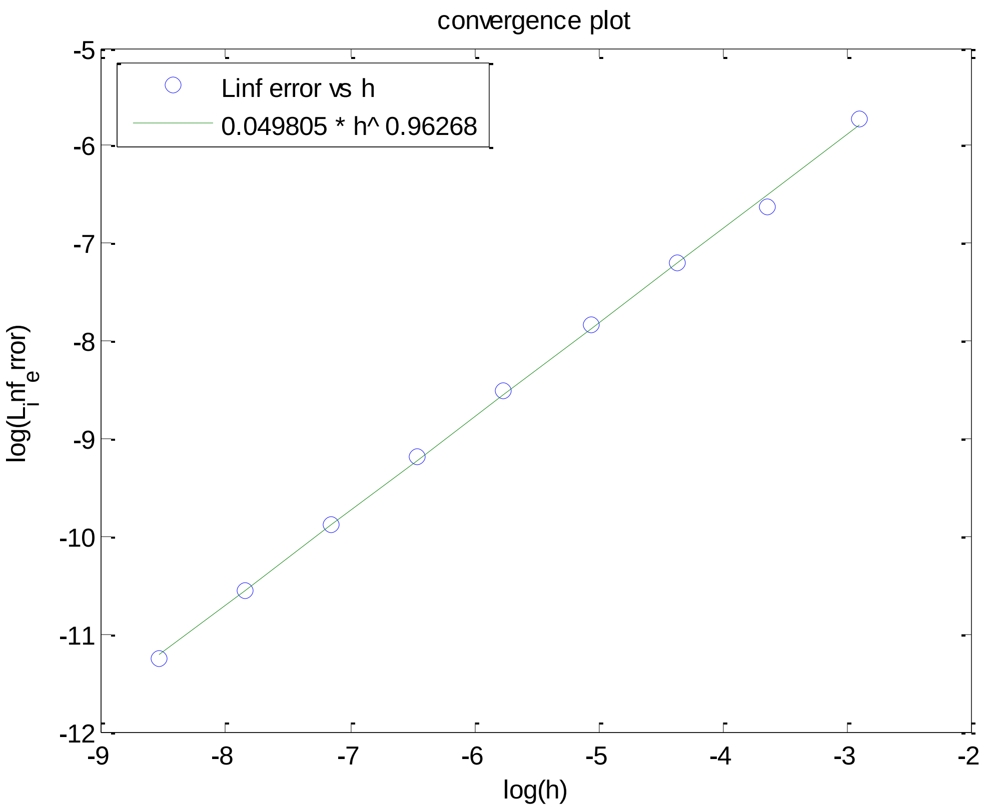
Use x_hat = 0.5. Use u(x) = sin(x) for x < x_hat and u(x) = e^x for x > x_hat as the exact solution.

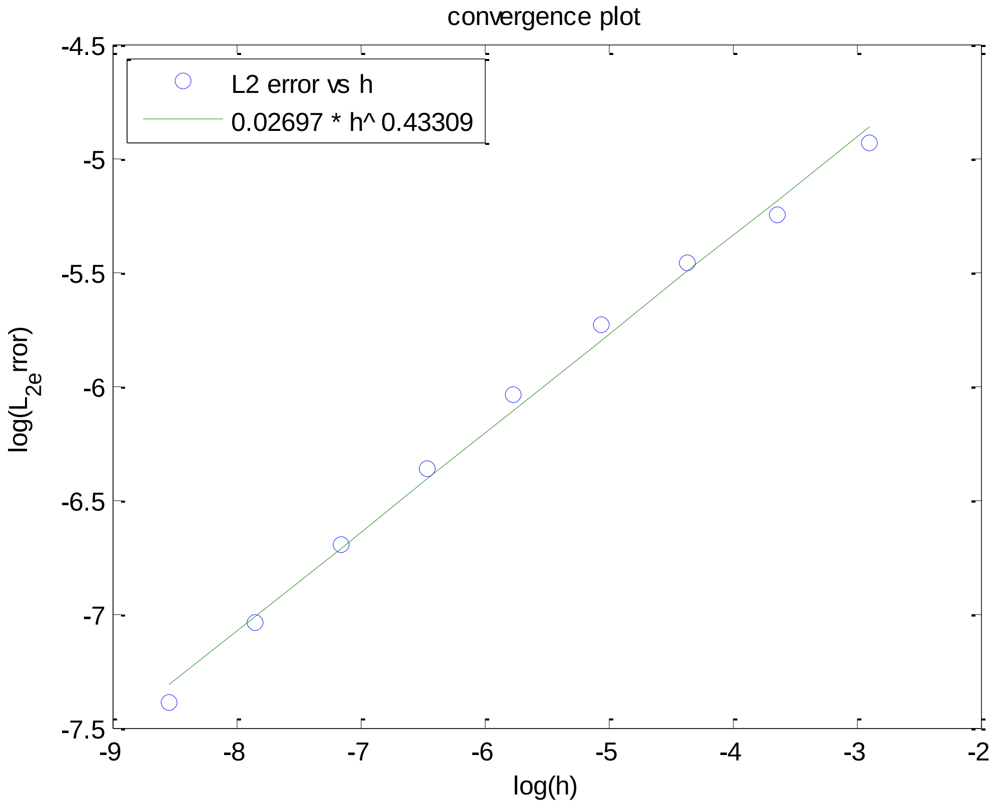
Since we have already derived the KKT system, the linear system is hard coded in Matlab.

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)');
```