```
close all;
clear all;

est1=[];
nvec=[];

elements=[1,3;3,2];
coordinates=[0;1;0.5];
nE=2;
theta=0.5;

while(nE<=500)
    nvec=[nvec,nE];
    % ToDo: Solve P1-FEM on coordinates, elements with output vector x

    % Plot Galerkin solution
    figure(2);
    [sc,idx]=sort(coordinates);
    plot(sc,x(idx),'-g');

    %Refine mesh uniformly
    %ToDo: Implement function refineMesh
    [coords_fine,els_fine,f2s]=refineMesh(coordinates,elements,1:nE);

    % ToDo: Solve P1-FEM on coords_fine, els_fine with output vector x_fine

    %ToDo: Compute H-H/2 error estimator
    est=computeHH2(coordinates, elements, f2s,x,x_fine);

    %Plot Error estimator
    est1=[est1,sqrt(sum(est))];
    figure(1);
    loglog(nvec,est1,'-*g',nvec,nvec.^(-1),'--k','LineWidth',2,'MarkerSize',5);
    hold on;
    drawnow;

    %DÃ¶rfler marking criterion
    [sest,idx]=sort(est);
    sest=sest(end:-1:1);
    sest=cumsum(sest);
    j=find(sest>=theta*sest(end),1,'first');
    marked=idx(end-j+1:end);

    %ToDo: Implement function refineMesh
    [coordinates,elements,~]=refineMesh(coordinates,elements,marked);
    nE=size(elements,1);
end
```