```
% project #4 in section 5.5 part c
rng(0) % to set the seed of random numbers

h=75; % interval width
a=201:h:876; % the left end point of each interval
b=275:h:950; % the right end point of each interval

% relative and cumulative percentages
rp=[2.5 2.5 5 7.5 12.5 17.5 42.5 5 2.5 2.5]/100';
cp=[2.5 5 10 17.5 30 47.5 90 95 97.5 100]/100'; % height
n=length(cp); % the number of intervals

% calculate the slopes
m(1)=cp(1)/(h/2); % first piece
m(n)=(cp(n)-cp(n-1)/(1.5*h)); % last piece
for k=2:n-1
    m(k)=(cp(k)-cp(k-1))/h;
end

% use the avg cost as the cost for that pool size
N=150000000; % number of simulations for each pool size S

Cost=[]; % a vector of costs at different
p=45; % pay rate for subs
r=81; % pay rate for regular overtime

Sk=1; % for the # of entries in cost
for S=590:1:610
    C=0; % cost for each S
    for k=1:N
        rd=rand; % each rand # will produce demand x by inverse splines
        % check class notes for the formula of the inverse spline
            if rd<cp(1), x0=(a(1)+b(1))/2; y0=cp(1); x=x0+(rd-y0)/m(1);
            elseif rd<cp(2), x0=(a(2)+b(2))/2; y0=cp(2); x=x0+(rd-y0)/m(2);
            elseif rd<cp(3), x0=(a(3)+b(3))/2; y0=cp(3); x=x0+(rd-y0)/m(3);
            elseif rd<cp(4), x0=(a(4)+b(4))/2; y0=cp(4); x=x0+(rd-y0)/m(4);
            elseif rd<cp(5), x0=(a(5)+b(5))/2; y0=cp(5); x=x0+(rd-y0)/m(5);
            elseif rd<cp(6), x0=(a(6)+b(6))/2; y0=cp(6); x=x0+(rd-y0)/m(6);
            elseif rd<cp(7), x0=(a(7)+b(7))/2; y0=cp(7); x=x0+(rd-y0)/m(7);
            elseif rd<cp(8), x0=(a(8)+b(8))/2; y0=cp(8); x=x0+(rd-y0)/m(8);
            elseif rd<cp(9), x0=(a(9)+b(9))/2; y0=cp(9); x=x0+(rd-y0)/m(9);
            else x0=b(10); y0=cp(10); x=x0+(rd-y0)/m(10);
            end
        if x<S, Cxs=p*S; % cost for that demand
        else Cxs=p*S+(x-S)*r;
        end
    C=C+Cxs; % add up the cost for each simulation
    end
    Cost(Sk)=C/N; % use the average as the cost for that S
    Sk=Sk+1;
end
plot(Cost,'*')
```