January 27, 2019

Empirircal Methods HA 6

Konstantin Guryev

Pennsylvania State University

2019

Problem 1

The Bellman equation is of the form:

$$V(p,x) = \max_{x' \in [0,x]} \{p(x-x') - \frac{1}{5}(x-x')^{\frac{3}{2}} + \delta E_{p'|p} V(p',x')\}$$

,

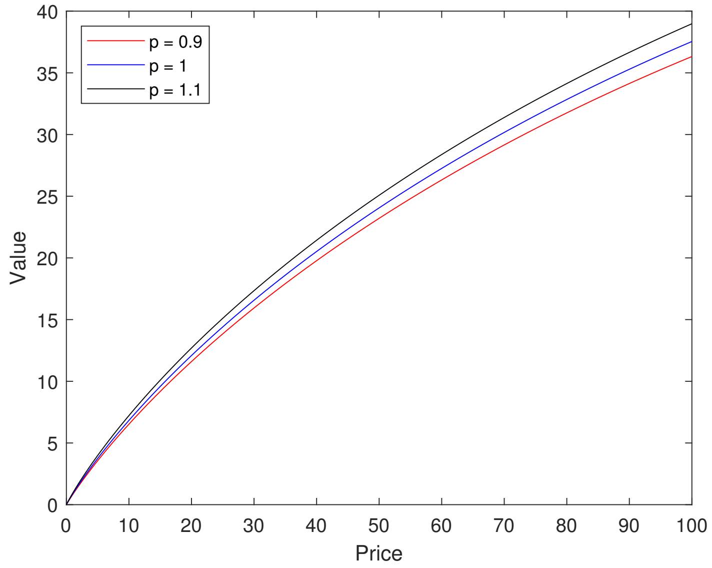
$$p' = \frac{1}{2} + \frac{1}{2}p + \epsilon, \delta = 0.95, x \in [0,100], \epsilon \sim N(0, 0.01)$$

A pair (p,x) is a state variable, where p is a price at a given time period and x is a stock of lumber in the firm. x' is a stock of lumber at the next period, which is our policy variable. Price follows an AR(1) process.

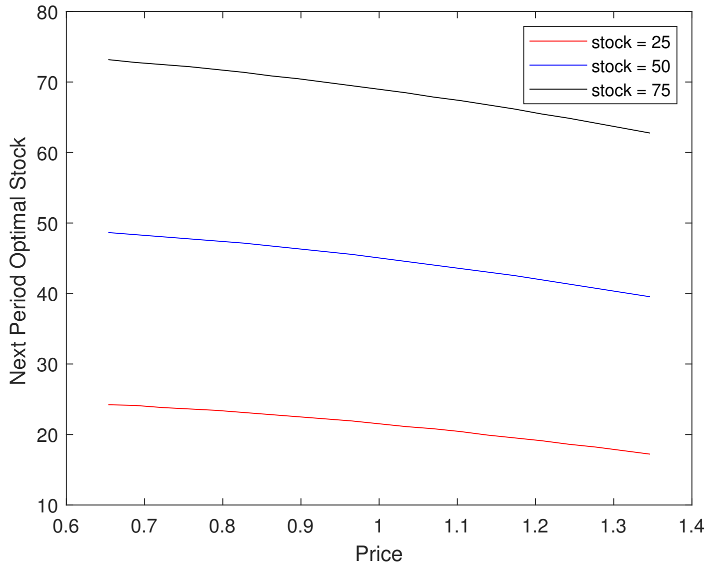
Problem 2

See the code.

Problem 3

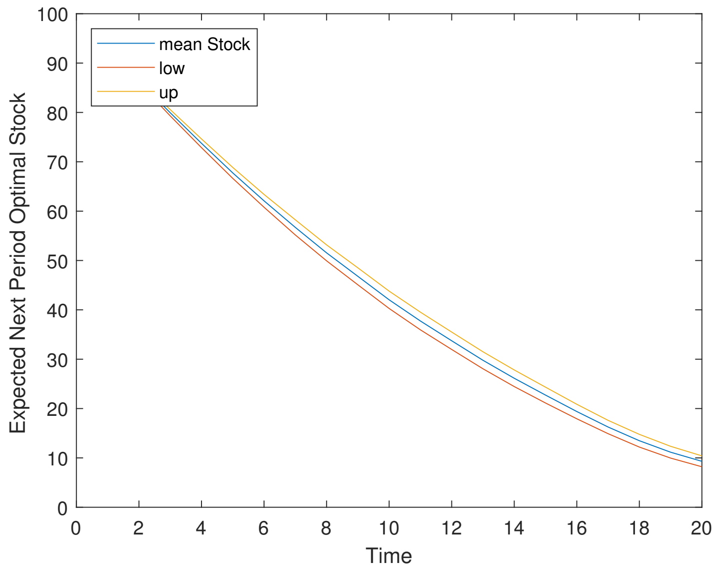


Problem 4



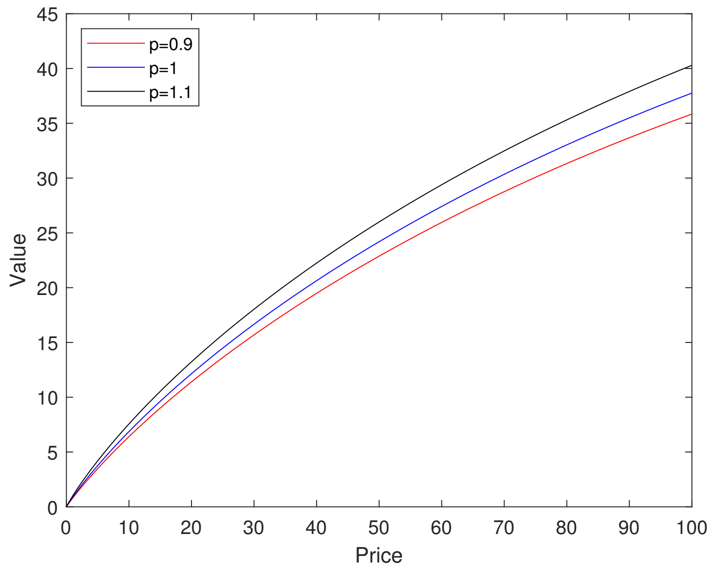
Problem 5

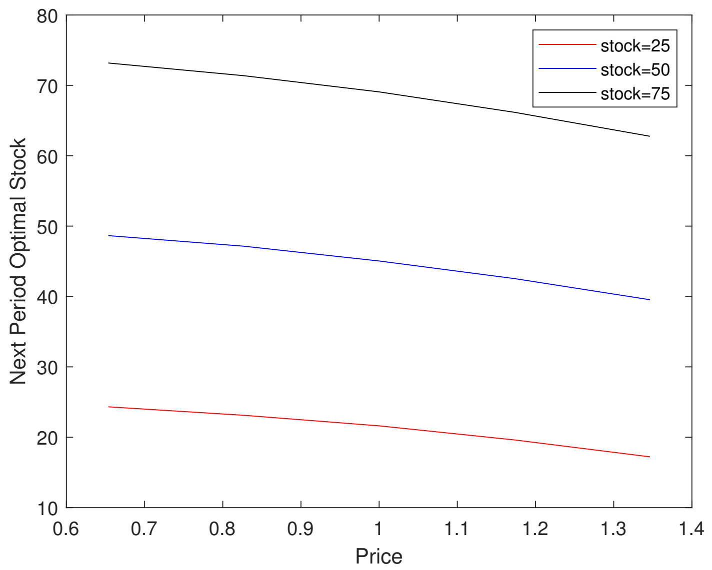
One may see that the confidence interval tends to expand over time. This result is natural since the further we are in the future the more uncertainty we face.



Problem 6

One may notice that on this grid the distance between the graphs of the value functions for the different prices is becoming larger when the price increases in comparison with the finer grid. Everything else seems to be pretty the same, therefore 5 points is also quite good a grid.





Matlab Code

```
function [prob,grid,invdist]=tauchen(N,mu,ro,sigma)

% N, number of grid points,
% mu, mean of the error term
% ro, AR(1) parameter
% sigma, std. of the error process


mux=mu/(1-ro);
sigmax=sigma/sqrt(1-ro^2);

epsl=mux-3*sigmax;          % this area (betw eps1 and epsl) is going to capture more than 99 per cent,
epsh=mux+3*sigmax;          % i.e. prob that an observation fall into this area( 3 std dev far from the mean) is at least 99%.
eps=linspace(epsl,epsh,N);
w=(eps(2)-eps(1))/2;    % half of the steps between the grids

if N==1; prob=1;
else

    p=zeros(N);
    p(:,1)=normcdf(((eps(1)+w-mu)*ones(N,1)-ro*eps')/sigma);      % prob that future shock will be eps(1), i.e.{P11, P21,..,Pn1}'
    p(:,N)=ones(N,1)-normcdf(((eps(N)-w-mu)*ones(N,1)-ro*eps')/sigma);     %the last column of the transition matrix
        for j=2:(N-1)
            p(:,j)=normcdf(((eps(j)+w-mu)*ones(N,1)-ro*eps')/sigma)-normcdf(((eps(j)-w-mu)*ones(N,1)-ro*eps')/sigma);
        end
end
    prob=p;
```

```
    grid=eps;


    Trans= prob';
probst = (1/N)*ones(N,1); % initial distribution of states
test = 1;

  while test > 10^(-8);
      probst1 = Trans*probst;
      test=max(abs(probst1-probst));
      probst = probst1;
   end
invdist=probst;
```

Matlab Code

```
function u = u(p,x)
x1 = x.*(x>=0);
out = p.*x1-0.2*x1.^(1.5);
u=out-100*(x<0);
end
```

Matlab Code

```
%% Problem 1
%See the pdf file

%% Problem 2

[prob,grid]=tauchen(21,0.5,0.5,0.1);
%% Problems 3 & 4

delta = 0.95;
S = 1000;
P = repmat(grid,S,1);
pi = prob;
stock = linspace(0,100,S);
S1 = repmat(stock',1,length(grid));
v_init = zeros(S,length(grid));d_init=ones(S,length(grid));
v_revised = ones(S,length(grid));d_revised=ones(S,length(grid));
diff=1;

while diff>0.001
    Ev=v_init*pi';
    for i=1:S
        U=u(P,stock(i)-S1);
        [v_revised(i,:),d_revised(i,:)]=max(U+delta*Ev);
    end
    diff=norm(v_revised-v_init)/norm(v_revised);
    disp(['Change in value function:',num2str(diff)]);
    v_init=v_revised;
end
policy = stock(d_revised);
figure(1)
plot(stock,v_init(:,8),'r-',stock,v_init(:,11),'b-',stock,v_init
    (:,14),'k-');
legend({'p = 0.9','p = 1','p = 1.1'},'Location','northwest');
xlabel('Price');
```

```
ylabel('Value');
figure(2)
plot(grid,policy(250,:),'r-',grid,policy(500,:),'b-',grid,policy
    (750,:),'k-');
legend({'stock = 25','stock = 50','stock = 75'},'Location','
    northeast');
xlabel('Price');
ylabel('Next Period Optimal Stock ');
%% Problem 5

 rng(1000);
 sims = 50;
 decision_sim = zeros(sims,20);
 p_gen = zeros(sims,20);

for k=1:sims
    % Generate the time series for price
    p_gen(k,1)=11; %vector of indices.
    for i=1:19
        draw=mnrnd(1,prob(p_gen(k,i),:));
        p_gen(k,i+1)=find(draw==1);
    end
    decision_sim(k,1)=d_revised(1000,11);
    for j=1:19
        decision_sim(k,j+1)=d_revised(decision_sim(k,j),p_gen(k,j+1)
            );
    end
    disp(['Iteration:', num2str(k)]);
end
sim_stock=stock(decision_sim);
sim_stock_mean=mean(sim_stock,1);
sim_stock_se=std(sim_stock,1);
sim_stock_lb=sim_stock_mean-1.645*(sim_stock_se)/sqrt(sims);
sim_stock_ub=sim_stock_mean+1.645*(sim_stock_se)/sqrt(sims);
plot(1:1:20,sim_stock_mean, 1:1:20,sim_stock_lb,1:1:20,sim_stock_ub
```

```
    );
legend({'mean Stock','low','up'},'Location','northwest');
xlabel('Time');
ylabel('Expected Next Period Optimal Stock ');

%% Problem 6

[prob1,grid1]=tauchen(5,0.5,0.5,0.1);

delta = 0.95;
S = 1000;
P = repmat(grid1,S,1);
pi2 = prob1;

stock1=linspace(0,100,S);
S1=repmat(stock1',1,length(grid1));
v_init2=zeros(S,length(grid1));d_init2=ones(S,length(grid1));
v_revised2=ones(S,length(grid1));d_revised2=ones(S,length(grid1));
diff2=1;

while diff2>0.001
    Ev=v_init2*pi2';
    for i=1:S
        U=u(P,stock1(i)-S1);
        [v_revised2(i,:),d_revised2(i,:)]=max(U+delta*Ev);
    end
    diff2=norm(v_revised2-v_init2)/norm(v_revised2);
    disp(['Change in value function:',num2str(diff2)]);
    v_init2=v_revised2;
end
policy2=stock1(d_revised2);

figure(3)
plot(stock1,v_init2(:,2),'r-',stock1,v_init2(:,3),'b-',stock1,
    v_init2(:,4),'k-');
```

```
legend({'p=0.9','p=1','p=1.1'},'Location','northwest');
xlabel('Price');
ylabel('Value');
figure(4)
plot(grid1,policy2(250,:),'r-',grid1,policy2(500,:),'b-',grid1,
    policy2(750,:),'k-');
legend({'stock=25','stock=50','stock=75'},'Location','northeast');
xlabel('Price');
ylabel('Next Period Optimal Stock');
```