Homework 6

Kensuke Suzuki

January 21, 2019

Problem 1: Dynamic optimization problem

Firm's current period payoff is

$$\pi(x_t, p_t) = p_t \cdot x_t - 0.2 \cdot x_t^{1.5} \tag{1}$$

We want to make the stock of lumber as a state variable, so that $x = k - k'$. Then equation (1) can be written as

$$\pi(k_t, k_{t+1}, p_t) = p \cdot (k_{t+1} - k_t) - 0.2 \cdot (k_{t+1} - k_t)^{1.5} \tag{2}$$

Then we can formulate the firm's problem such that it chooses k' (next period stock) given the state variables (k, p). The Bellman's equation is

$$V(k,p) = \max_{k'} p \cdot (k' - k) - 0.2 \cdot (k' - k)^{1.5} + \delta \mathbb{E}_{p'} \left[V(k', p') | p \right] \tag{3}$$

given initial stock of lumber (between 0 and 100) and transition probability of p which is defined by the AR(1) process ($p' = p_0 + \rho \cdot p + u$). In the Bellman's equation presented in equation (2), expectation is taken over next period price p' given today's price p.

Problem 2: Tauchen

I use `tauchen.m` to generate grid that approximate process for p_t for given AR(1) parameters. Variable `grid` stores the grid points for p_t and `prob` stores the transition probabilities.

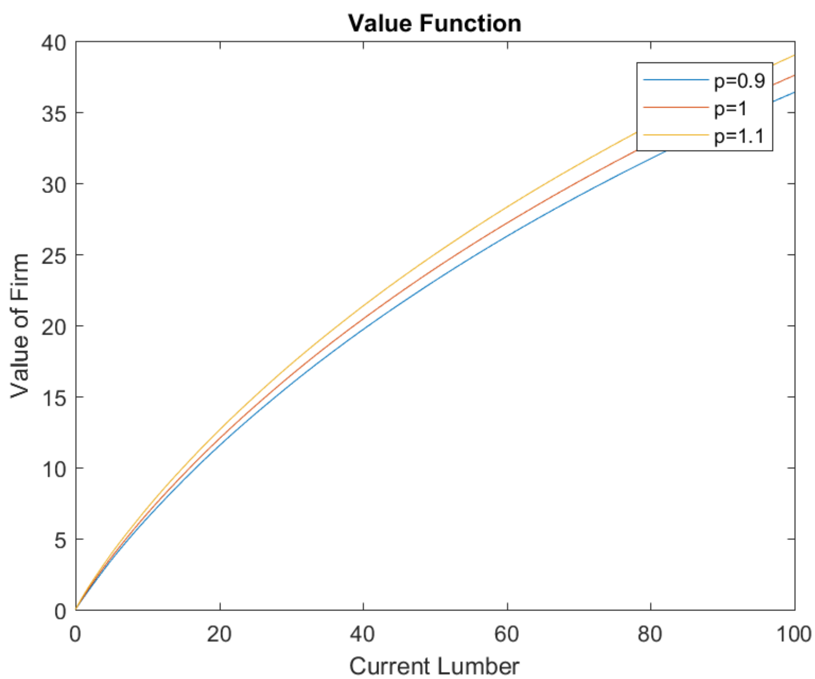
Problem 3: Value function iteration (VIF)

The basic structure of the VIF algorithm is analogous to the one provided in the lecture. Note that:

- `harvest` stores the harvest for given current stock (in column) and next period stock (in raw)
- I penalize the negative revenue by replacing with $-1e6.0$
- In order to avoid imaginary number, before calculating the current payoff (profit), I replace the negative harvest with 0.

I plot the value of the firm depending on its initial stock (x-axis) and the current price of lumber.

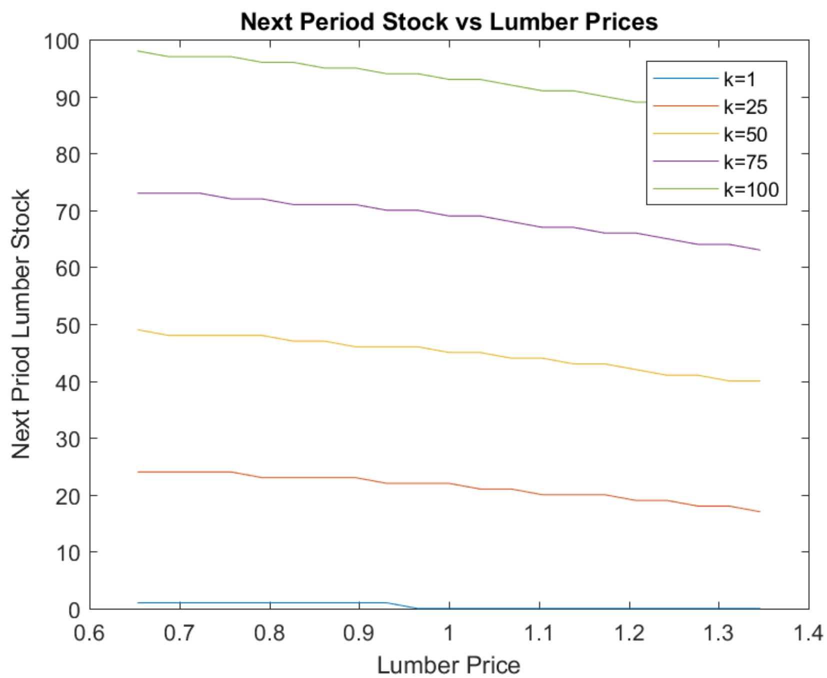
Figure 1: Problem 3



Problem 4: Next period stock as a function of price

I plot next period optimal stock as a function of today's price for different amount of lumber left in stock.

Figure 2: Problem 4

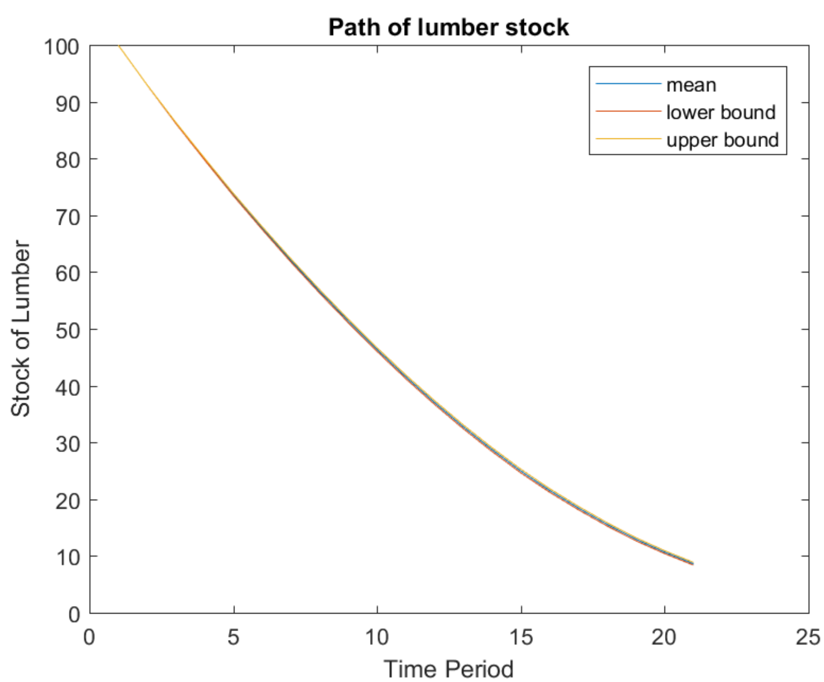


Problem 5: Expected stock

I compute the expected path in the following way.

1. Simulate the path of price for 20 period (p_t for $t \in [1, 20]$), given $p_0 = 1$, as follows.
   - Using the transition probability matrix, compute the *cumulative probability* for given current period price (p_t). `cdf` is the vector where ith element stores $cdf_i = \sum_{j=1}^{i} = \pi(p_j|p_t)$.
   - Draw a uniform random variable $r \in [0, 1]$ using `rand` and find the smallest index i such that $cdf_i > r$. Then set the next priod price as $p_{t+1} = p_i$.
2. Using the path of price obtained in Step 1, we can compute the path of lumber stock. Next period lumber stock is determined by the decision rule which is obtained in Problem 3 above. Denote the path of lumber stock obtained in this step by k_t^s for $t = 1, 2, ..., 20$. s is the index of the simulation, which is explained below.
3. Repeat the Step 1–2 for $S = 1000$ times, $s = 1, 2, ..., S$. Expected lumber stock is computed as the sample average of k_t^s: $\overline{k}_t = \frac{1}{S}\sum_{s=1}^{S} k_t^s$.
   Confidence interval is obtained by
   $$\overline{k}_t - t_{0.05}(S-1)\frac{SD_t}{\sqrt{S}} < \mathbb{E}\left[k_t\right] < \overline{k}_t + t_{0.05}(S-1)\frac{SD_t}{\sqrt{S}}$$
   where $t_{0.05}(S-1)$ and $t_{0.95}(S-1)$, respectively, are 5%tile and 95%tile of t-distribution with degree of freedom $S-1$. std_t is the standard deviation of k_t^s: $SD_t = \sqrt{\frac{1}{S-1}\sum_{s=1}^{S}\left|k_t^s - \overline{k}_t\right|}$.

Figure 3: Problem 5



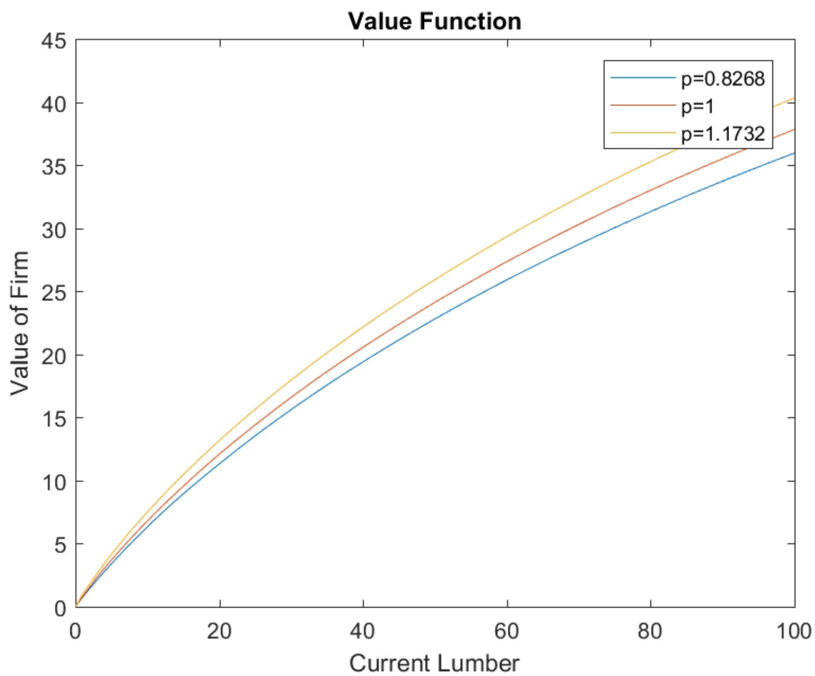
Problem 6: Results with coarse grid

Problem 6-2: Tauchen

I know use 5 grid points and generate the grid and transition probability for p_t.

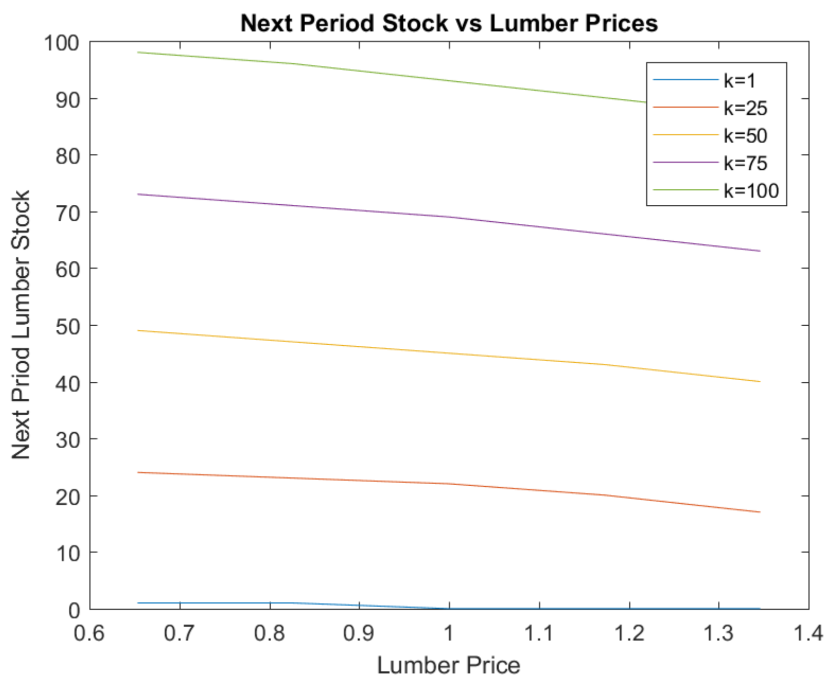
Problem 6-3: VIF

Figure 4: Problem 6-3



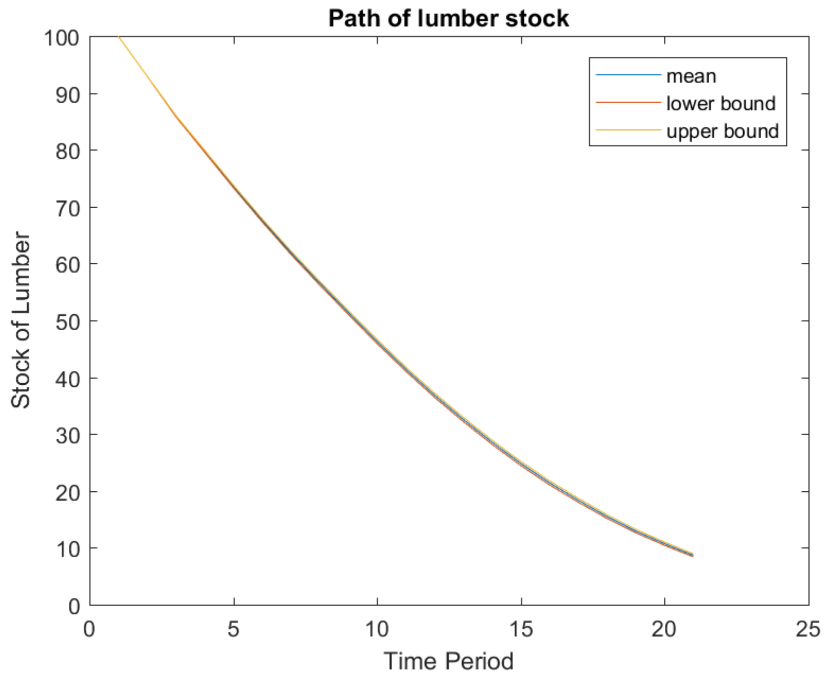
Problem 6-4: Next period stock as a function of price

Figure 5: Problem 6-4



Problem 6-5: Expected stock

Figure 6: Problem 6-5



Matlab Main Code

```
% Spring 2019
% ECON512 Empirical Method
% Homework 6 --- Dynamic Programing
% Kensuke Suzuki
% kxs974@psu.edu

clear all;

%% Basic setup
delta = 0.95;
p0 = 0.5;
rho = 0.5;
sigma = 0.1;

N = 101 ;% number of grid
k = linspace(0,100,N);

%% Problem 1
disp('Refer to the PDF')

%% Problem 2 --- Tauchen

Z = 21; % grid points generated

[prob,grid]=tauchen(Z,p0,rho,sigma);
disp(['The dimensions of prob are ' num2str(size(prob)) ])
disp(['The dimensions of grid are ' num2str(size(grid)) ])


%% Problem 3 --- Value function iteration

vinitial=zeros(N,Z);
vrevised=zeros(N,Z);
decision=zeros(N,Z);

invest=kron(ones(1,Z),k');
disp(['The dimensions of harvest  are ' num2str(size(invest)) ])
ONE=ones(N,1);

diff = 1;
while diff > 1e-6

    Ev = vinitial*prob' ;% (NxZ * ZxZ = NxZ);

    Ev;

    for i = 1:N
        harvest = ( kron(ones(N,Z),k(:,i)') - invest) ; % harvest
```

```
        revenue = (ones(N,1)*grid) .* harvest; % revenue
        revenue(revenue < 0) = -1e6; % punish negative revenue
        harvest = max(harvest, 0); % replace negative harvest to avoide
            imaginary num
        pi = revenue - 0.2 * (harvest.^(1.5)); % current payoff
        [vrevised(i,:),decision(i,:)] = max( pi + delta*Ev ); % get value
            and policy function
        decision;
    end
    diff = norm(vrevised - vinitial) / norm(vrevised); % check deviance
    vinitial = vrevised;

end

plot(k,vrevised(:,8)',k,vrevised(:,11)',k,vrevised(:,14)')
xlabel('Current Lumber')
ylabel('Value of Firm')
title('Value Function')
legend('p=0.9','p=1', 'p=1.1')
saveas(gcf,'prob3.png')

%
% %%
%
% plot(grid, decision(2,:), grid, decision(26,:),grid, decision(51,:),grid,
    decision(76,:),grid, decision(101,:))
% title('Next Period Stock as a Function of Price')
% xlabel('Price')
% ylabel('Next period stock of lumber')


%% Problem 4

drule=zeros(N,Z);

for i=1:Z
    drule(:,i)=k(decision(:,i))'; % get decision rule
end

plot(grid, drule(2,:), grid, drule(26,:),grid, drule(51,:),grid, drule
    (76,:),grid, drule(101,:))
title('Next Period Stock vs Lumber Prices')
xlabel('Lumber Price')
ylabel('Next Priod Lumber Stock')
legend('k=1','k=25', 'k=50', 'k=75', 'k=100')
saveas(gcf,'prob4.png')


%% Problem 5 Expected path

```

```
T = 21; % time period
Time = 1:T;
sim = 1000; % number of simulation

path_p = zeros(sim,T);
path_p(:,1) = 11;

% generate the price path (20 period)  1000 times
for s = 1:1000
    for i = 2:21
        rng(s * 2019 + i );
        pmf = prob(path_p(s,i-1),:);
        cdf = cumsum(pmf);
        r = rand;
        path_p(s,i) = find(cdf>r,1);
    end
end

% generate the lumber stock path
path_k = zeros(sim,T);
path_k(:,1) = 101; % this is index
path_stock = zeros(sim,T);
path_stock(:,1) = 100;

for s = 1:1000
    for i = 2:21
        path_k(s,i) = decision(path_k(s,i-1), path_p(s,i-1));
        path_stock(s,i) = k(path_k(s,i));
    end
end

mean = mean(path_stock);
stderr = std(path_stock);
ts = tinv([0.05  0.95],length(path_stock-1))';

CI = ones(2,1)*mean + ts*stderr/sqrt(length(path_stock)) ;

plot(Time, mean, Time, CI(1,:), Time, CI(2,:))
title('Path of lumber stock')
xlabel('Time Period')
ylabel('Stock of Lumber')
legend('mean', 'lower bound', 'upper bound')
saveas(gcf,'prob5.png')


%% Problem 6 Tauchen with coarse grid

clear prob grid vinitial vrevused decision mean

Z = 5; % grid points generated
```

```
[prob,grid]=tauchen(Z,p0,rho,sigma);
disp(['The dimensions of prob are ' num2str(size(prob)) ])
disp(['The dimensions of grid are ' num2str(size(grid)) ])

vinitial=zeros(N,Z);
vrevised=zeros(N,Z);
decision=zeros(N,Z);

invest=kron(ones(1,Z),k');
disp(['The dimensions of harvest  are ' num2str(size(invest)) ])
ONE=ones(N,1);

vinitial=zeros(N,Z);
vrevised=zeros(N,Z);
decision=zeros(N,Z);

invest=kron(ones(1,Z),k');
disp(['The dimensions of harvest  are ' num2str(size(invest)) ])
ONE=ones(N,1);

diff = 1;
while diff > 1e-6

    Ev = vinitial*prob' ;% (NxZ * ZxZ = NxZ);

    Ev;

    for i = 1:N
        harvest = ( kron(ones(N,Z),k(:,i)') - invest) ;
        revenue = (ones(N,1)*grid) .* harvest;
        revenue(revenue < 0) = -1e6;
        harvest = max(harvest, 0);
        pi = revenue - 0.2 * (harvest.^(1.5));

        %pi = (ones(N,1)*grid) .* harvest - 0.2 * (harvest_z.^(1.5));
        check = pi + delta*Ev;
        [vrevised(i,:),decision(i,:)] = max( pi + delta*Ev );
        decision;
    end
    diff = norm(vrevised - vinitial) / norm(vrevised);
    vinitial = vrevised;

end

plot(k,vrevised(:,2)',k,vrevised(:,3)',k,vrevised(:,4)')
xlabel('Current Lumber')
ylabel('Value of Firm')
title('Value Function')
legend('p=0.8268','p=1', 'p=1.1732')
```

```
saveas(gcf,'prob6_1.png')


drule=zeros(N,Z);

for i=1:Z
    drule(:,i)=k(decision(:,i))';
end


plot(grid, drule(2,:), grid, drule(26,:),grid, drule(51,:),grid, drule
    (76,:),grid, drule(101,:))
title('Next Period Stock vs Lumber Prices')
xlabel('Lumber Price')
ylabel('Next Priod Lumber Stock')
legend('k=1','k=25', 'k=50', 'k=75', 'k=100')
saveas(gcf,'prob6_2.png')

T = 21; % time period
Time = 1:T;
sim = 1000; % number of simulation

path_p = zeros(sim,T);
path_p(:,1) = 3;

% generate the price path (20 period)  1000 times
for s = 1:1000
    for i = 2:21
        rng(s * 2019 + i );
        pmf = prob(path_p(s,i-1),:);
        cdf = cumsum(pmf);
        r = rand;
        path_p(s,i) = find(cdf>r,1);
    end
end

% generate the lumber stock path
path_k = zeros(sim,T);
path_k(:,1) = 101; % this is index
path_stock = zeros(sim,T);
path_stock(:,1) = 100;

for s = 1:1000
    for i = 2:21
        path_k(s,i) = decision(path_k(s,i-1), path_p(s,i-1));
        path_stock(s,i) = k(path_k(s,i));
    end
end

mean = mean(path_stock);
```

```
stderr = std(path_stock);
ts = tinv([0.05  0.95],length(path_stock-1))';

CI = ones(2,1)*mean + ts*stderr/sqrt(length(path_stock)) ;

plot(Time, mean, Time, CI(1,:), Time, CI(2,:))
title('Path of lumber stock')
xlabel('Time Period')
ylabel('Stock of Lumber')
legend('mean', 'lower bound', 'upper bound')
saveas(gcf,'prob6_3.png')
```