

## Homework 6

---

Kensuke Suzuki

January 21, 2019

### Problem 1: Dynamic optimization problem

Firm's current period payoff is

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

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

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

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

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

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.

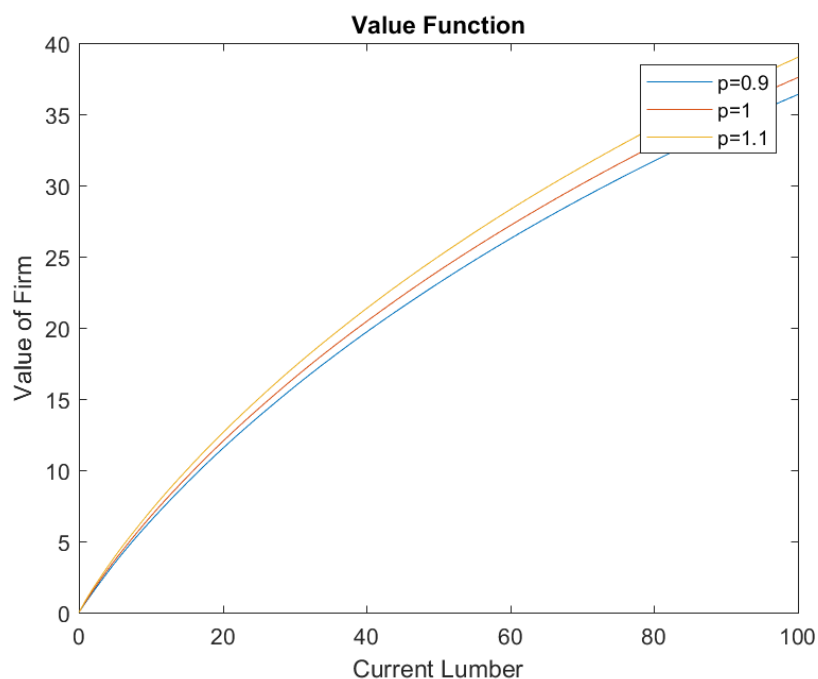
### Problem 3: Value function iteration (VIF)

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 row)
- 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.

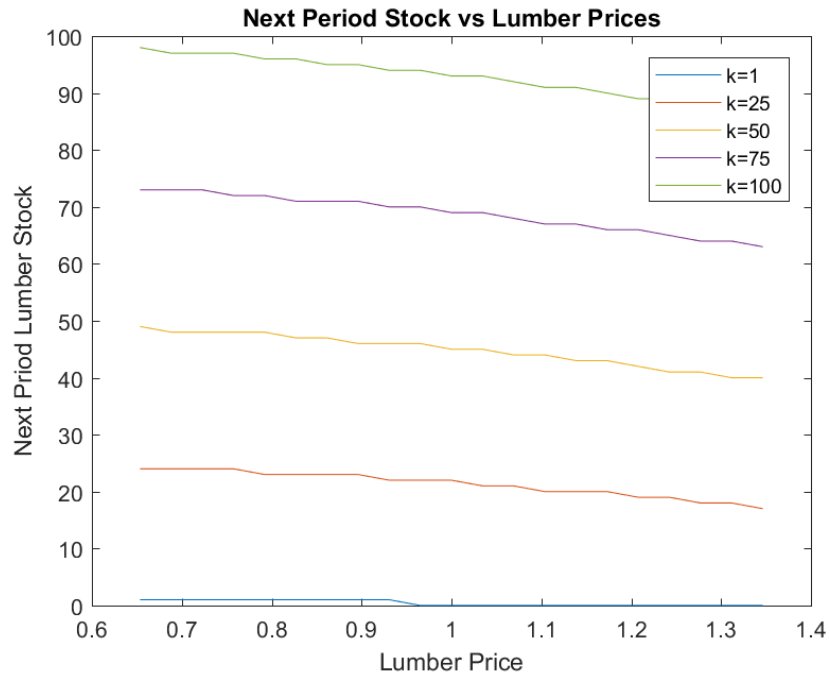
Figure 1: Problem 3



### 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.

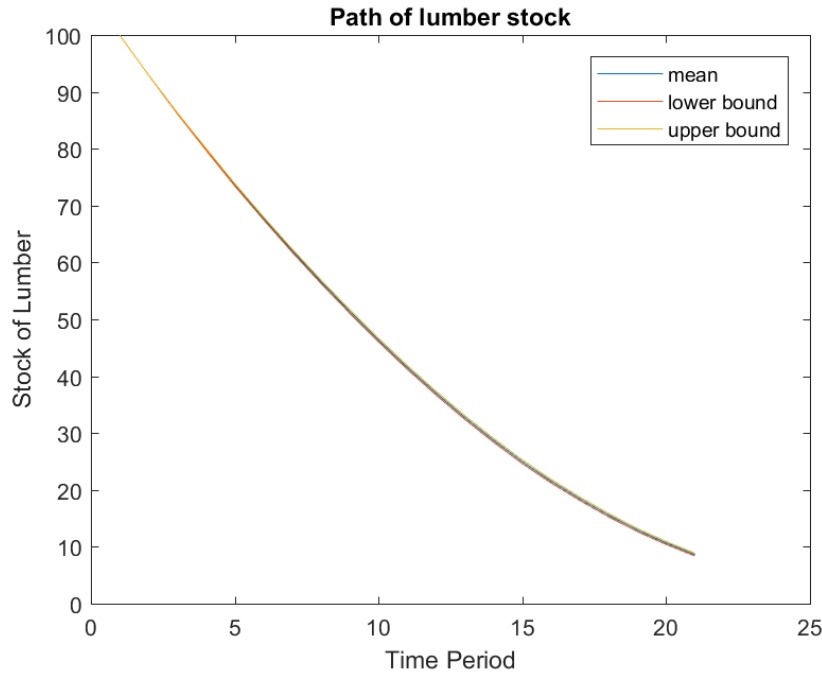
1. Simulate the path of price for 20 period ( $p_t$  for  $t \in [1, 20]$ ), given  $p_0 = 1$ , as follows.
  - Using the transition probability matrix, compute the *cumulative probability* for given current period price ( $p_t$ ).  $cdf$  is the vector where  $i$ th element stores  $cdf_i = \sum_{j=1}^i \pi(p_j|p_t)$ .
  - 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 period 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  $t = 1, 2, \dots, 20$ .  $s$  is the index of the simulation, which is explained below.
3. Repeat the Step 1–2 for  $S = 1000$  times,  $s = 1, 2, \dots, S$ . Expected lumber stock is computed as the sample average of  $k_t^s$ :  $\bar{k}_t = \frac{1}{S} \sum_{s=1}^S k_t^s$ .

Confidence interval is obtained by

$$\bar{k}_t - t_{0.05}(S-1) \frac{SD_t}{\sqrt{S}} < \mathbb{E}[k_t] < \bar{k}_t + t_{0.05}(S-1) \frac{SD_t}{\sqrt{S}}$$

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  $k_t^s$ :  $SD_t = \sqrt{\frac{1}{S-1} \sum_{s=1}^S |k_t^s - \bar{k}_t|}$ .

Figure 3: Problem 5



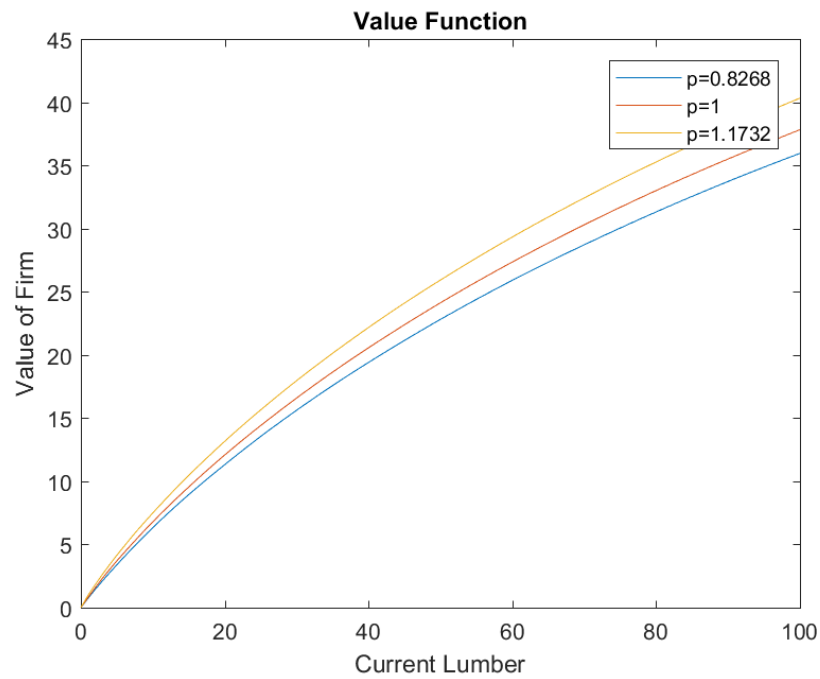
## Problem 6: Results with coarse grid

### Problem 6-2: Tauchen

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

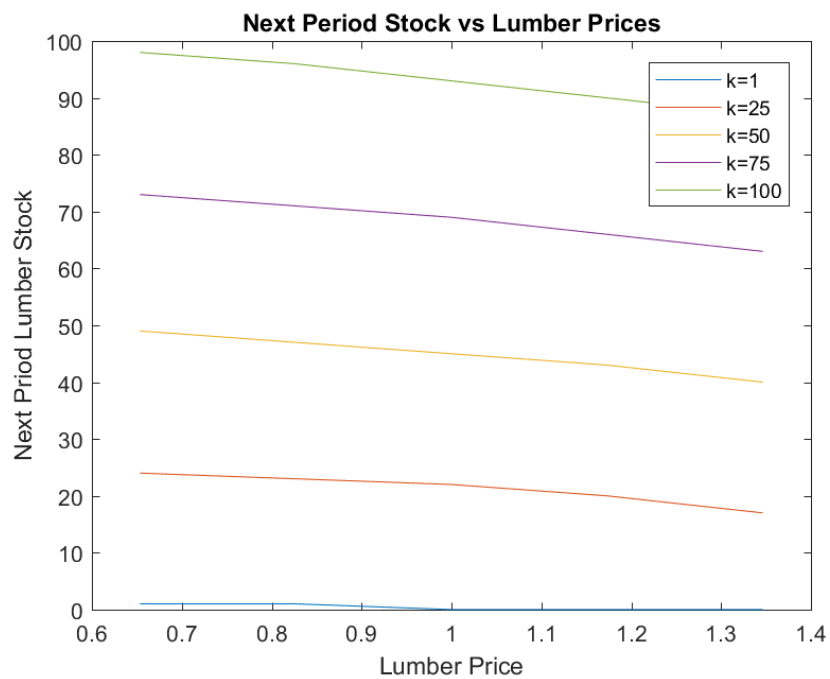
### Problem 6-3: VIF

Figure 4: Problem 6-3



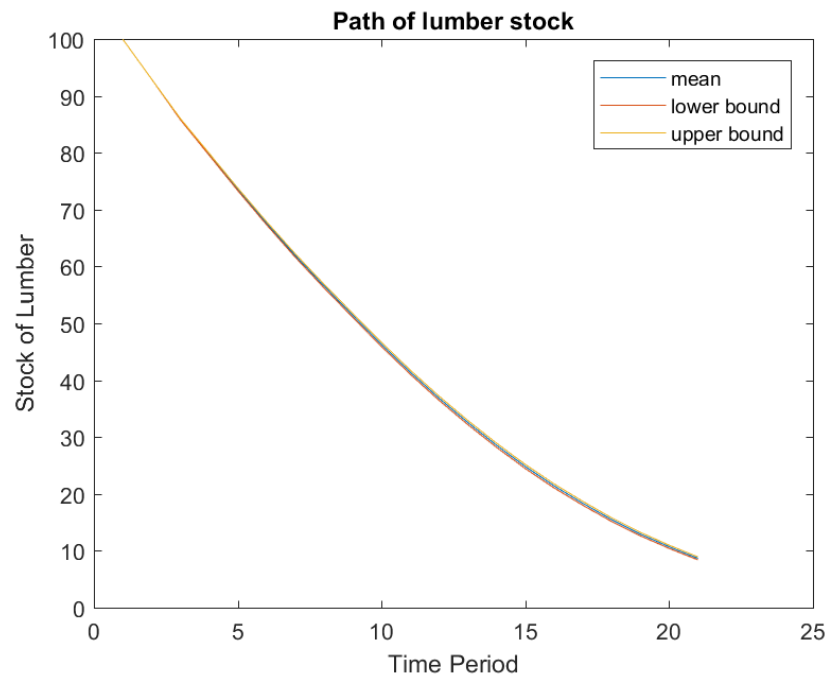
**Problem 6-4: Next period stock as a function of price**

Figure 5: Problem 6-4



**Problem 6-5: Expected stock**

Figure 6: Problem 6-5



## Matlab Main Code

```
1 % Spring 2019
2 % ECON512 Empirical Method
3 % Homework 6 — Dynamic Programming
4 % Kensuke Suzuki
5 % kxs974@psu.edu
6
7 clear all;
8
9 %% Basic setup
10 delta = 0.95;
11 p0 = 0.5;
12 rho = 0.5;
13 sigma = 0.1;
14
15 N = 101 ;% number of grid
16 k = linspace(0,100,N);
17
18 %% Problem 1
19 disp('Refer to the PDF')
20
21 %% Problem 2 — Tauchen
22
23 Z = 21; % grid points generated
24
25 [prob, grid]=tauchen(Z,p0,rho,sigma);
26 disp(['The dimensions of prob are ' num2str(size(prob)) ])
27 disp(['The dimensions of grid are ' num2str(size(grid)) ])
28
29
30 %% Problem 3 — Value function iteration
31
32 vinitial=zeros(N,Z);
33 vrevised=zeros(N,Z);
34 decision=zeros(N,Z);
35
36 invest=kron(ones(1,Z),k');
37 disp(['The dimensions of harvest are ' num2str(size(invest)) ])
38 ONE=ones(N,1);
39
40 diff = 1;
41 while diff > 1e-6
42
43     Ev = vinitial*prob' ;% (NxZ * ZxZ = NxZ);
44
45     Ev;
46
47     for i = 1:N
48         harvest = ( kron(ones(N,Z),k(:,i)') - invest) ; % harvest
```



```

49     revenue = (ones(N,1)*grid) .* harvest; % revenue
50     revenue(revenue < 0) = -1e6; % punish negative revenue
51     harvest = max(harvest, 0); % replace negative harvest to avoid
        imaginary num
52     pi = revenue - 0.2 * (harvest.^(1.5)); % current payoff
53     [vrevise(i,:),decision(i,:)] = max( pi + delta*Ev ); % get value
        and policy function
54     decision;
55 end
56     diff = norm(vrevise - vinitial) / norm(vrevise); % check deviance
57     vinitial = vrevise;
58
59 end
60
61 plot(k,vrevise(:,8)',k,vrevise(:,11)',k,vrevise(:,14)')
62 xlabel('Current Lumber')
63 ylabel('Value of Firm')
64 title('Value Function')
65 legend('p=0.9','p=1','p=1.1')
66 saveas(gcf,'prob3.png')
67
68 %
69 % %%
70 %
71 % plot(grid, decision(2,:), grid, decision(26,:),grid, decision(51,:),grid,
        decision(76,:),grid, decision(101,:))
72 % title('Next Period Stock as a Function of Price')
73 % xlabel('Price')
74 % ylabel('Next period stock of lumber')
75
76
77 %% Problem 4
78
79 drule=zeros(N,Z);
80
81 for i=1:Z
82     drule(:,i)=k(decision(:,i)); % get decision rule
83 end
84
85 plot(grid, drule(2,:), grid, drule(26,:),grid, drule(51,:),grid, drule
        (76,:),grid, drule(101,:))
86 title('Next Period Stock vs Lumber Prices')
87 xlabel('Lumber Price')
88 ylabel('Next Priod Lumber Stock')
89 legend('k=1','k=25','k=50','k=75','k=100')
90 saveas(gcf,'prob4.png')
91
92
93 %% Problem 5 Expected path
94

```

```

95 T = 21; % time period
96 Time = 1:T;
97 sim = 1000; % number of simulation
98
99 path_p = zeros(sim,T);
100 path_p(:,1) = 11;
101
102 % generate the price path (20 period) 1000 times
103 for s = 1:1000
104     for i = 2:21
105         rng(s * 2019 + i );
106         pmf = prob(path_p(s,i-1),:);
107         cdf = cumsum(pmf);
108         r = rand;
109         path_p(s,i) = find(cdf>r,1);
110     end
111 end
112
113 % generate the lumber stock path
114 path_k = zeros(sim,T);
115 path_k(:,1) = 101; % this is index
116 path_stock = zeros(sim,T);
117 path_stock(:,1) = 100;
118
119 for s = 1:1000
120     for i = 2:21
121         path_k(s,i) = decision(path_k(s,i-1), path_p(s,i-1));
122         path_stock(s,i) = k(path_k(s,i));
123     end
124 end
125
126 mean = mean(path_stock);
127 stderr = std(path_stock);
128 ts = tinv([0.05 0.95],length(path_stock)-1)';
129
130 CI = ones(2,1)*mean + ts*stderr/sqrt(length(path_stock)) ;
131
132 plot(Time, mean, Time, CI(1,:), Time, CI(2,:))
133 title('Path of lumber stock')
134 xlabel('Time Period')
135 ylabel('Stock of Lumber')
136 legend('mean', 'lower bound', 'upper bound')
137 saveas(gcf,'prob5.png')
138
139
140 %% Problem 6 Tauchen with coarse grid
141
142 clear prob grid vinitial vrevused decision mean
143
144 Z = 5; % grid points generated

```

```

145
146 [prob, grid]=tauchen(Z,p0,rho,sigma);
147 disp(['The dimensions of prob are ' num2str(size(prob)) ])
148 disp(['The dimensions of grid are ' num2str(size(grid)) ])
149
150 vinitial=zeros(N,Z);
151 vrevised=zeros(N,Z);
152 decision=zeros(N,Z);
153
154 invest=kron(ones(1,Z),k');
155 disp(['The dimensions of harvest are ' num2str(size(invest)) ])
156 ONE=ones(N,1);
157
158 vinitial=zeros(N,Z);
159 vrevised=zeros(N,Z);
160 decision=zeros(N,Z);
161
162 invest=kron(ones(1,Z),k');
163 disp(['The dimensions of harvest are ' num2str(size(invest)) ])
164 ONE=ones(N,1);
165
166 diff = 1;
167 while diff > 1e-6
168
169     Ev = vinitial*prob' ;% (NxZ * ZxZ = NxZ);
170
171     Ev;
172
173     for i = 1:N
174         harvest = ( kron(ones(N,Z),k(:,i)') - invest) ;
175         revenue = (ones(N,1)*grid) .* harvest;
176         revenue(revenue < 0) = -1e6;
177         harvest = max(harvest, 0);
178         pi = revenue - 0.2 * (harvest.^(1.5));
179
180         %pi = (ones(N,1)*grid) .* harvest - 0.2 * (harvest_z.^(1.5));
181         check = pi + delta*Ev;
182         [vrevised(i,:),decision(i,:)] = max( pi + delta*Ev );
183         decision;
184     end
185     diff = norm(vrevised - vinitial) / norm(vrevised);
186     vinitial = vrevised;
187
188 end
189
190 plot(k,vrevised(:,2)',k,vrevised(:,3)',k,vrevised(:,4)')
191 xlabel('Current Lumber')
192 ylabel('Value of Firm')
193 title('Value Function')
194 legend('p=0.8268','p=1','p=1.1732')

```

```
195 saveas(gcf, 'prob6_1.png')
196
197
198 drule=zeros(N,Z);
199
200 for i=1:Z
201     drule(:,i)=k(decision(:,i));
202 end
203
204
205 plot(grid, drule(2,:), grid, drule(26,:),grid, drule(51,:),grid, drule
      (76,:),grid, drule(101,:))
206 title('Next Period Stock vs Lumber Prices')
207 xlabel('Lumber Price')
208 ylabel('Next Priod Lumber Stock')
209 legend('k=1', 'k=25', 'k=50', 'k=75', 'k=100')
210 saveas(gcf, 'prob6_2.png')
211
212 T = 21; % time period
213 Time = 1:T;
214 sim = 1000; % number of simulation
215
216 path_p = zeros(sim,T);
217 path_p(:,1) = 3;
218
219 % generate the price path (20 period) 1000 times
220 for s = 1:1000
221     for i = 2:21
222         rng(s * 2019 + i);
223         pmf = prob(path_p(s,i-1),:);
224         cdf = cumsum(pmf);
225         r = rand;
226         path_p(s,i) = find(cdf>r,1);
227     end
228 end
229
230 % generate the lumber stock path
231 path_k = zeros(sim,T);
232 path_k(:,1) = 101; % this is index
233 path_stock = zeros(sim,T);
234 path_stock(:,1) = 100;
235
236 for s = 1:1000
237     for i = 2:21
238         path_k(s,i) = decision(path_k(s,i-1), path_p(s,i-1));
239         path_stock(s,i) = k(path_k(s,i));
240     end
241 end
242
243 mean = mean(path_stock);
```

```
244 stderr = std(path_stock);
245 ts = tinv([0.05 0.95],length(path_stock-1))';
246
247 CI = ones(2,1)*mean + ts*stderr/sqrt(length(path_stock)) ;
248
249 plot(Time, mean, Time, CI(1,:), Time, CI(2,:))
250 title('Path of lumber stock')
251 xlabel('Time Period')
252 ylabel('Stock of Lumber')
253 legend('mean', 'lower bound', 'upper bound')
254 saveas(gcf, 'prob6_3.png')
```