

ECON 6140 - Problem Set # 3

Julien Manuel Neves

April 19, 2018

Open Sector Growth Model via Dynamic Programming

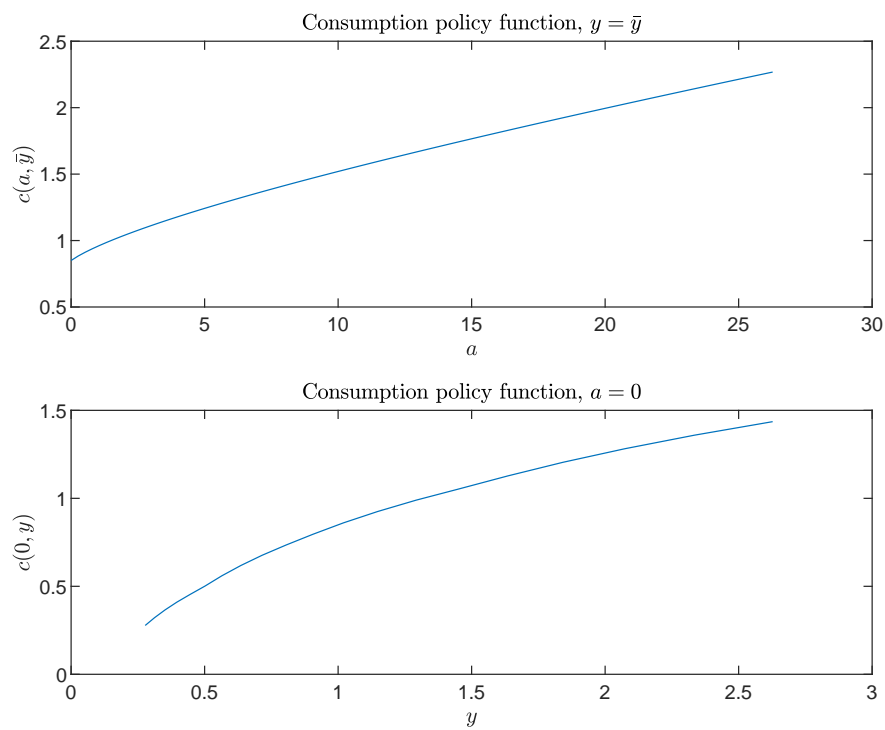


Figure 1: Value function

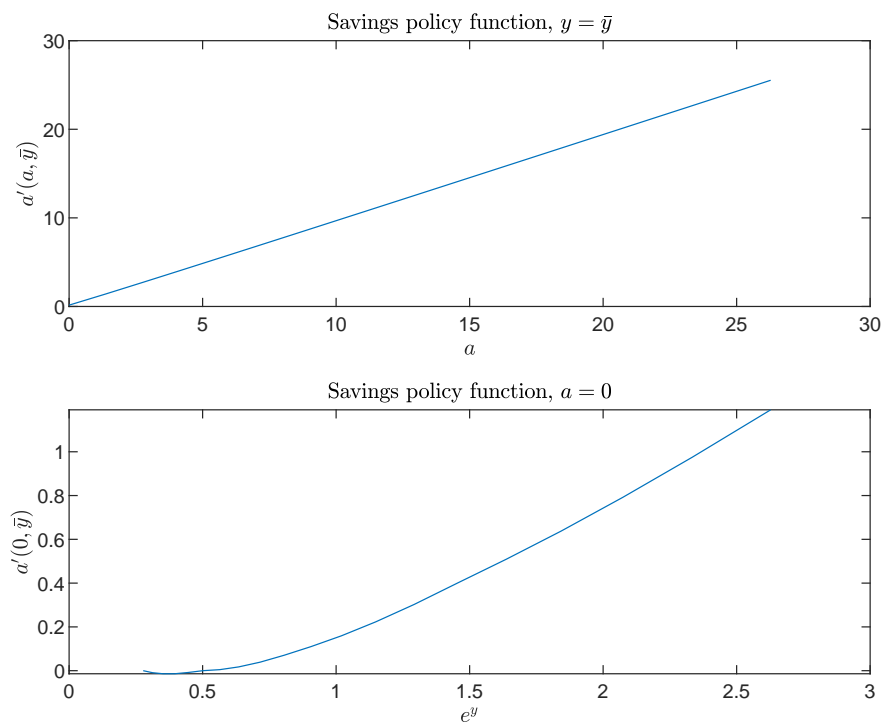


Figure 2: Policy function

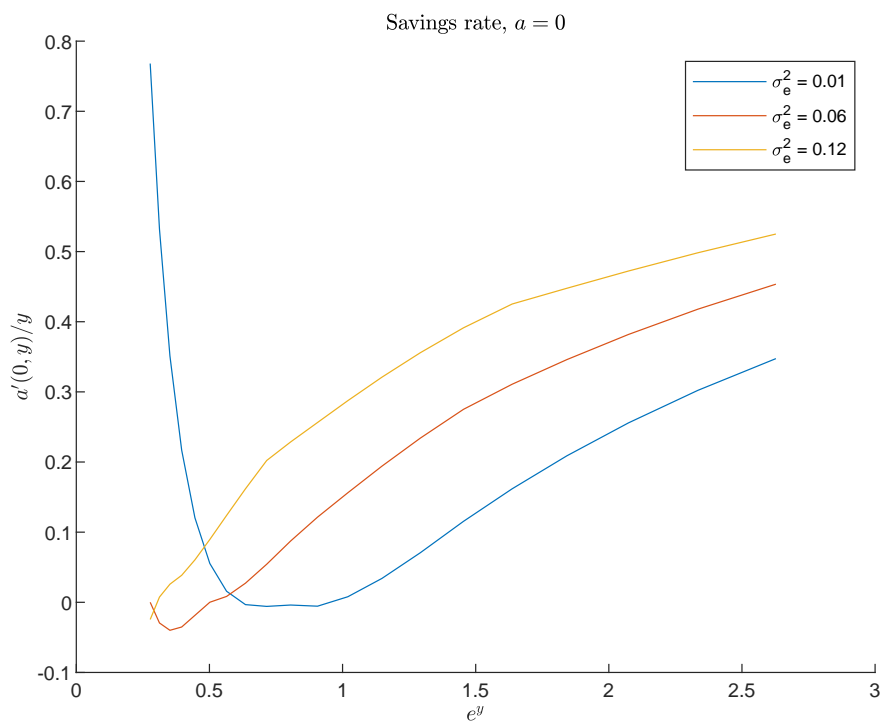


Figure 3: Time path for consumption and capital

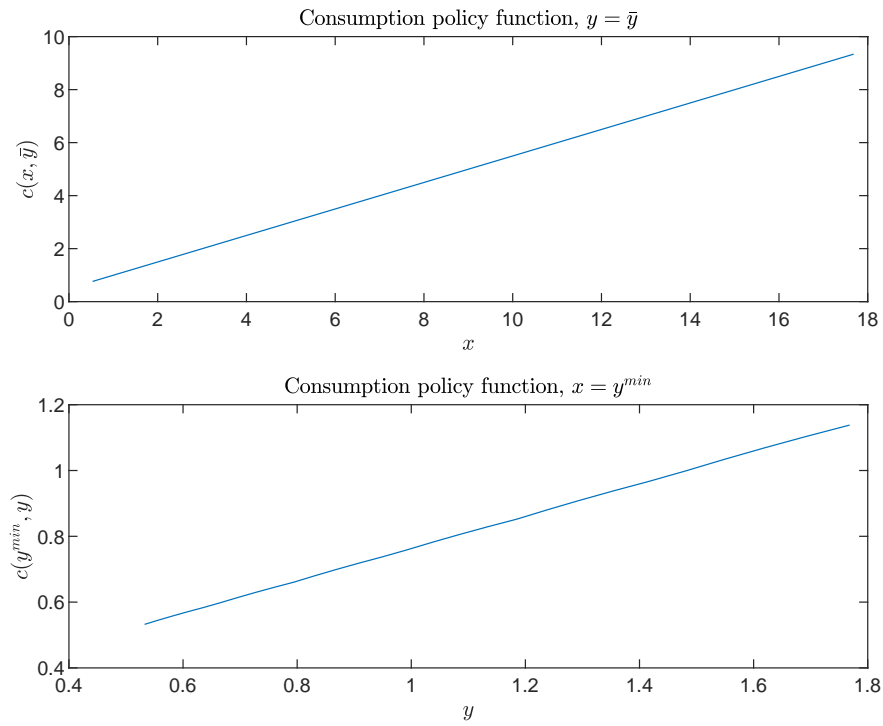


Figure 4: Time path for consumption and capital

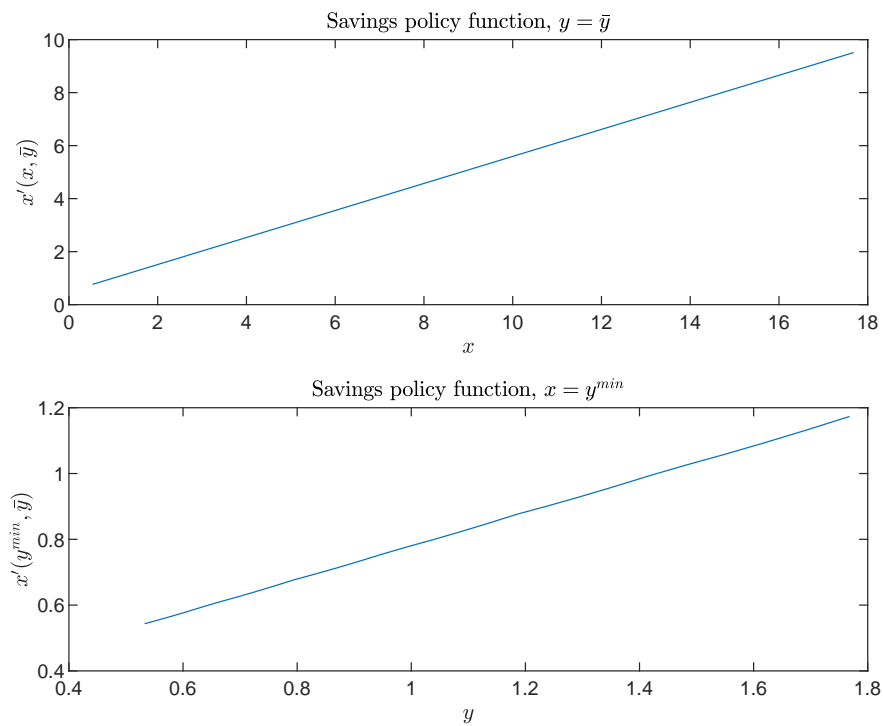


Figure 5: Time path for consumption and capital

Code

```
1 clear; clc;
2 %% Problem 1
3 % declare parameters
4 param.ro=0.90;
5 param.se=sqrt(0.06);
6
7 % part a) - b)
8 param.k = 5;
9 [param, ro_ta5, se_ta5] = transition(param, "tauchen");
10 param.k = 10;
11 [param, ro_ta10, se_ta10] = transition(param, "tauchen");
12
13 tab_1b = table([param.ro; ro_ta5; ro_ta10],[param.se; se_ta5;
    se_ta10], 'VariableNames',{ 'rho', 'se' }, 'RowNames',{ 'Model'; 'Five'; 'Ten' })
14
15 % part c)
16 param.k = 5;
17 [param, ro_rw, se_rw] = transition(param, "rouwenhorst");
18
19 tab_1c = table([param.ro; ro_ta5; ro_rw],[param.se; se_ta5; se_rw],
    'VariableNames',{ 'rho', 'se' }, 'RowNames',{ 'Model'; 'tauchen'; 'rouwenhorst' })
20
21 % part d)
22 old_var = param.se^2/(1-param.ro^2);
23 param.ro = 0.98;
24 param.se = old_var*(1-param.ro^2);
25 [param, ro_rw, se_rw] = transition(param, "rouwenhorst");
26
27 tab_1d = table([param.ro; ro_rw],[param.se; se_rw], 'VariableNames',{
    'rho', 'se' }, 'RowNames',{ 'Model'; 'rouwenhorst' })
28
29 %% Problem 2
30 % declare parameters
31 param.beta=0.95;
32 param.r=0.02;
33 param.ro=0.9;
34 param.se=sqrt(0.06);
35 param.gamma=2;
36 param.amin = 0;
37 param.n = 25;
```

```

38 param.k = 5;
39
40 % part a)
41 [param] = transition(param, "rouwenhorst");
42 [param,c,fspace,s,smin,smax] = policy_ip(param);
43
44 close all
45 sfine=gridmake(nodeunif(param.n*2,smin(1),smax(1)),param.ygrid);
46 xfine=funeval(c,fspace,sfine);
47
48 figure(1)
49 subplot(2,1,1)
50 sfine=gridmake(nodeunif(param.n*4,smin(1),smax(1)),0); %ygrid(
    floor(k/2)+2));
51 xfine=funeval(c,fspace,sfine);
52 plot(sfine(:,1),xfine)
53 xlabel({'$a$'},'Interpreter','latex')
54 ylabel({'$c(a,\bar{y})$'},'Interpreter','latex')
55 title({'Consumption policy function, $y=\bar{y}$'},'Interpreter','
    latex')
56 set(gca,'FontSize',8);
57
58 subplot(2,1,2)
59 sfine=gridmake(0,nodeunif(param.k*4,smin(2),smax(2)));
60 xfine=funeval(c,fspace,sfine);
61 plot(exp(sfine(:,2)),xfine)
62 xlabel({'$y$'},'Interpreter','latex')
63 ylabel({'$c(0,y)$'},'Interpreter','latex')
64 title({'Consumption policy function, $a=0$'},'Interpreter','latex'
    )
65 set(gca,'FontSize',8);
66 print -depsc fig1.eps
67
68 figure(2)
69 subplot(2,1,1)
70 sfine=gridmake(nodeunif(param.k*4,smin(1),smax(1)),0);
71 xfine=funeval(c,fspace,sfine);
72 plot(sfine(:,1),(1+param.r)*sfine(:,1)+exp(sfine(:,2))-xfine)
73 xlabel({'$a$'},'Interpreter','latex')
74 ylabel({'$a^{\backslash prime}(a,\bar{y})$'},'Interpreter','latex')
75 title({'Savings policy function, $y=\bar{y}$'},'Interpreter','
    latex')
76 set(gca,'FontSize',8);
77
78 subplot(2,1,2)

```

```

79 sfine=gridmake(0,nodeunif(param.k*4,smin(2),smax(2)));
80 xfine=funeval(c,fspace,sfine);
81 plot(exp(sfine(:,2)),(1+param.r)*sfine(:,1)+exp(sfine(:,2))-xfine)
82 xlabel('$e^y$', 'Interpreter', 'latex')
83 ylabel('$a^{\prime}(0,\bar{y})$', 'Interpreter', 'latex')
84 title({'Savings policy function, $a=0$'}, 'Interpreter', 'latex')
85 set(gca, 'FontSize', 8);
86 print -depsc fig2.eps
87
88 % part b)
89 gamma = [1, 2, 5];
90 for i = 1:3
91     param.gamma = gamma(i);
92
93     [param] = transition(param, "rouwenhorst");
94     [param, c, fspace] = policy_ip(param);
95
96     con = markovchain(param, c, fspace, 10000); % Generate Markov
          chain
97     se_c(i) = std(con);
98 end
99
100 tab_2b = table(se_c, 'VariableNames', {'std_c'}, 'RowNames', {'Gamma
          = 1'; 'Gamma = 2'; 'Gamma = 5'})
101
102
103 % part c)
104 param.gamma = 2;
105 se = [0.01, 0.06, 0.12];
106
107 figure(3)
108 hold on
109 for i = 1:3
110     param.se = sqrt(se(i));
111
112     [param] = transition(param, "rouwenhorst");
113     [param, c, fspace] = policy_ip(param);
114
115     sfine=gridmake(0,nodeunif(param.k*4,smin(2),smax(2)));
116     xfine=funeval(c,fspace,sfine);
117     plot(exp(sfine(:,2)), 1-xfine./exp(sfine(:,2)))
118 end
119 xlabel('$e^y$', 'Interpreter', 'latex')
120 ylabel('$a^{\prime}(0,y)/y$', 'Interpreter', 'latex')
121 title({'Savings rate, $a=0$'}, 'Interpreter', 'latex')

```

```

122 legend( '\sigma_e^2 = 0.01 ', '\sigma_e^2 = 0.06 ', '\sigma_e^2 = 0.12 '
123 )
124 set(gca, 'FontSize', 8);
125 hold off
126 print -depsc fig3.eps
127
128 % part d) - e)
129 % no-borrowing
130 param.gamma = 2;
131 param.se = sqrt(0.06);
132 param.amin = 0;
133
134 [param] = transition(param, "rouwenhorst");
135 [param,c,fspace] = policy_ip(param);
136
137 [con, s] = markovchain(param,c,fspace, 10000);
138 avg_c(1) = mean(con);
139
140 c_t = log(con(2:end))-log(con(1:end-1));
141 e_t = s(2:end,2)-param.ro*s(1:end-1,2);
142 sig = cov(c_t,e_t);
143 phi(1) = 1-sig(1,2)/sig(2,2);
144
145 % natural debt limit
146 param.amin = -min(exp(param.ygrid)+.01)/param.r;
147
148 [param] = transition(param, "rouwenhorst");
149 [param,c,fspace] = policy_ip(param);
150
151 [con, s] = markovchain(param,c,fspace, 10000);
152 avg_c(2) = mean(con);
153
154 c_t = log(con(2:end))-log(con(1:end-1));
155 e_t = s(2:end,2)-param.ro*s(1:end-1,2);
156 sig = cov(c_t,e_t);
157 phi(2) = 1-sig(1,2)/sig(2,2);
158
159 tab_2d = table(avg_c', phi', 'VariableNames', {'mean_c', 'phi'}, '
    RowNames', {'No borrowing'; 'Natural'})
160
161
162 %% Problem 3
163 % declare parameters
164 param.beta=0.95;

```

```

165 param.r=0.02;
166 param.ro=0;
167 param.se=sqrt(0.06);
168 param.gamma=2;
169 param.amin = 0;
170 param.n = 25;
171 param.k = 7;
172
173 % part b)
174 wbar = -param.se^2/2;
175 [x,w] = qwnorm(7,wbar,param.se^2);
176 fprintf('E(y)=%f \n',exp(x)*w)
177
178 % part c)
179 %param.ws = w';
180 %param.ygrid = x;
181 [param] = transition(param, "rouwenhorst");
182 [param,c,fspace,s,smin,smax] = policy_ca(param);
183
184 sfine=gridmake(nodeunif(param.n(1)*2,smin(1),smax(1)),param.ygrid)
185 ;
186 xfine=funeval(c,fspace,sfine);
187
188 figure(4)
189 subplot(2,1,1)
190 sfine=gridmake(nodeunif(param.n*4,smin(1),smax(1)),0); %ygrid(
191 floor(k/2)+2));
192 xfine=funeval(c,fspace,sfine);
193 plot(sfine(:,1),xfine)
194 xlabel({'$x$'},'Interpreter','latex')
195 ylabel({'$c(x,\bar{y})$'},'Interpreter','latex')
196 title({'Consumption policy function, $y=\bar{y}$'},'Interpreter','
197 latex')
198 set(gca,'FontSize',8);
199
200 subplot(2,1,2)
201 sfine=gridmake(smin(1),nodeunif(param.k*4,smin(2),smax(2)));
202 xfine=funeval(c,fspace,sfine);
203 plot(exp(sfine(:,2)),xfine)
204 xlabel('$y$', 'Interpreter', 'latex')
205 ylabel('$c(y^{\min},y)$', 'Interpreter', 'latex')
206 title({'Consumption policy function, $x=y^{\min}$'}, 'Interpreter', '
207 latex')
208 set(gca,'FontSize',8);
209 print -depsc fig4.eps

```



```

206
207 figure(5)
208 subplot(2,1,1)
209 sfine=gridmake(nodeunif(param.k*4,smin(1),smax(1)),0);
210 xfine=funeval(c,fspace,sfine);
211 plot(sfine(:,1),(1+param.r)*(sfine(:,1)-xfine)+exp(sfine(:,2)))
212 xlabel('$x$', 'Interpreter', 'latex')
213 ylabel('$x^{\prime}(x,\bar{y})$', 'Interpreter', 'latex')
214 title({'Savings policy function, $y=\bar{y}$'}, 'Interpreter', '
    latex')
215 set(gca, 'FontSize', 8);
216
217 subplot(2,1,2)
218 sfine=gridmake(smin(1),nodeunif(param.k*4,smin(2),smax(2)));
219 xfine=funeval(c,fspace,sfine);
220 plot(exp(sfine(:,2)),(1+param.r)*sfine(:,1)+exp(sfine(:,2))-xfine)
221 xlabel('$y$', 'Interpreter', 'latex')
222 ylabel('$x^{\prime}(y^{\min},\bar{y})$', 'Interpreter', 'latex')
223 title({'Savings policy function, $x=y^{\min}$'}, 'Interpreter', '
    latex')
224 set(gca, 'FontSize', 8);
225 print -depsc fig5.eps

```