February 19, 2019

Empirical Methods HA 7

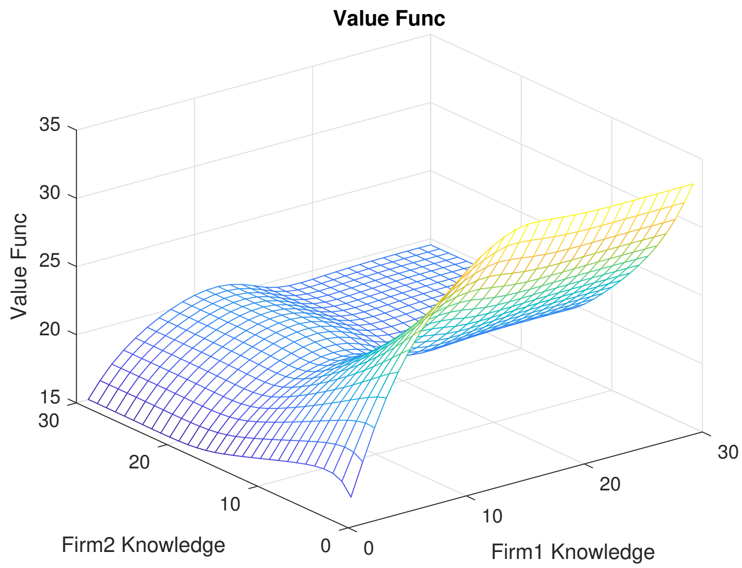
Konstantin Guryev

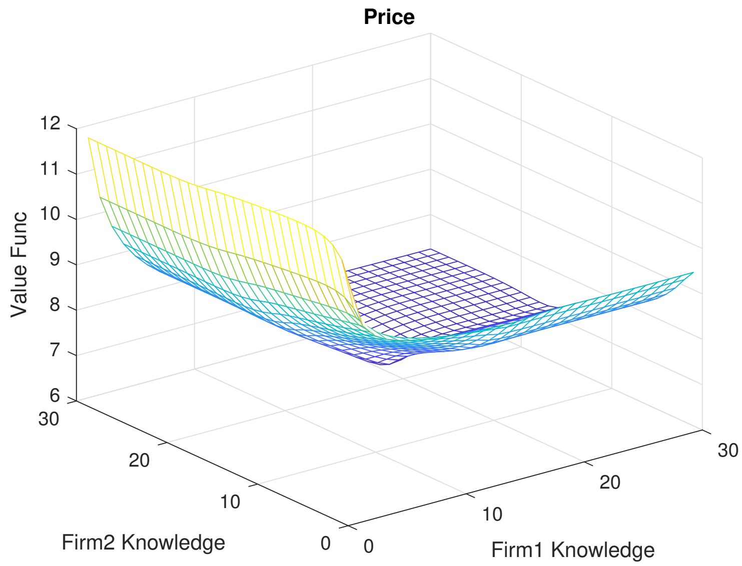
Pennsylvania State University

2019

Question 1

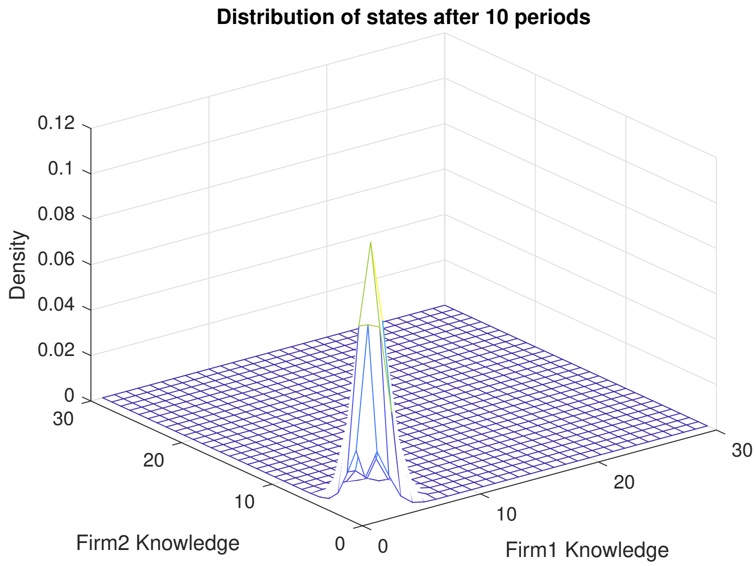
The proposed algorithm converges within 303 seconds and on the 82^{th} iteration for $\lambda = 1$, i.e. in the case when the dampening is not used. In case when $\lambda = 0.5$ the number of iterations is twice larger, namely 165, and the time of convergence is 598 seconds.

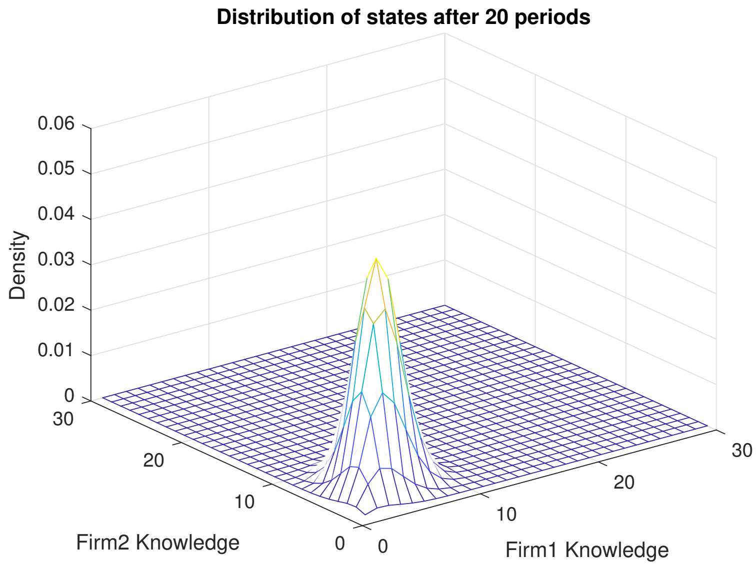


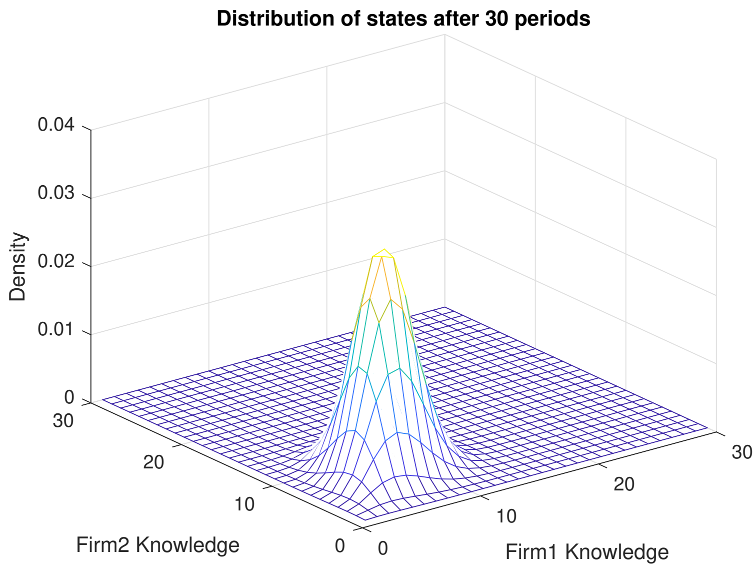


Question 2

To tackle a problem a 900 by 900 matrix is constructed in which each row contains the probability to going to one of any 900 consequent periods.

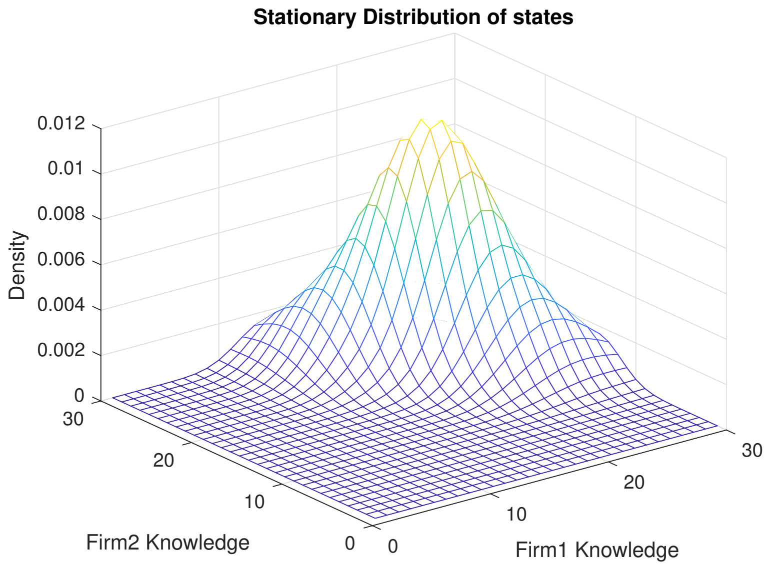






Question 3

To obtain the steady state distribution I just raised the matrix to the power of 900 (relatively big number) and reshaped the rows to be 30 by 30.



Matlab Code

```
global  l L c v delta rho beta kappa stop lambda;

l = 15;
L = 30;
c = zeros(L,1);
v = 10;
delta = 0.03;
rho = 0.85;
beta = 1/1.05;
kappa = 10;

stop = 1e-3;
lambda = 1;

%I also tried lambda = 0.5 as proposed in dampening procedure. The number of iterations to converge was twice
%larger

c(1:l)=kappa*[1:1:l]'.^(log(rho)/log(2)); %Constructing cost function
c(l+1:end) = kappa*(l^(log(rho)/log(2)));
tr_pr=zeros(L+1,L); %Constructing the state transition conditional
```

```
    on sales
for i=1:L+1
    if i==1
        tr_pr(i,1)=1;
    elseif i==L+1
            tr_pr(i,L)=1;
    else
        tr_pr(i,i-1)=1-(1-delta)^(i);
        tr_pr(i,i)=(1-delta)^(i);
    end
end
```

Matlab Code

```
function [d0,d1,d2] = Ds(p,p_1,v)
d0 = 1./(1+exp(v-p)+exp(v-p_1'));
d1 = exp(v-p)./(1+exp(v-p)+exp(v-p_1'));
d2 = exp(v-p_1')./(1+exp(v-p)+exp(v-p_1'));
end
```

Matlab Code

```
function [w0,w1,w2]=Ws(V,tr_pr)

s = size(tr_pr,1);
t_p = tr_pr;
tr_pr_0 = t_p(1:s-1,:);
tr_pr_1 = t_p(2:s,:);

w0 = tr_pr_0*V*tr_pr_0';
w1 = tr_pr_1*V*tr_pr_0';
w2=tr_pr_0*V*tr_pr_1';

end
```

Matlab Code

```
function f = FOC(V,p,p_1,L,c,beta,v,tr_pr)
c1 = repmat(c',L,1);
[d0, d1, d2] = Ds(p,p_1,v);
[w0, w1,w2] = Ws(V,tr_pr);
FOC=1-(1-d1).*(p-c1)-beta*w1+beta*(d0.*w0+d1.*w1+d2.*w2);
f = FOC;
end
```

Matlab Code

```matlab
function [v,p,it] = VFI(V,P,stop,L,c,lambda,beta,v,tr_pr)

v_old = V;
p_old = P;
initial = P;
opt = optimset('Disp','None');
v_new = zeros(size(V));
p_new = zeros(size(P));
tol = 1;
i = 1;

while and(tol > stop, i < 1000)
    f=@(p)FOC(v_old,p,p_old,L,c,beta,v,tr_pr);
    p_new = fsolve(f,initial,opt);
    [d0, d1, d2] = Ds(p_new,p_old,v);
    [w0, w1, w2] = Ws(v_old,tr_pr);

    v_new = d1.*(p_new-repmat(c',L,1))+beta*(d0.*w0+d1.*w1+d2.*w2);

    tol = max(max(max(abs((v_new-v_old)./(1+v_new)))),max(max(abs((p_new-p_old)./(1+p_new)))));

    v_old=lambda*v_new+(1-lambda)*v_old; p_old=lambda*p_new+(1-lambda)*p_old;

    fprintf('Iteration:%d\n',i);
    fprintf('Tolerance:%f\n',tol);
    i=i+1;
end
v = v_old;
p = p_old;
it = i;
end
```

Matlab Code

```
%% Question 1

tic;
Parameters;
V_0 = (repmat(v,L,L) + repmat(c',L,1))./(2*beta);
P_0 = 1.5*repmat(c',L,1);

[value, p, it] = VFI(V_0, P_0, stop, L, c, lambda, beta, v, tr_pr);

figure(1);
mesh(value);
title('Value Func');
xlabel('Firm1 Knowledge');
ylabel('Firm2 Knowledge');
zlabel('Value Func');
figure(2);
mesh(p);
title('Price');
xlabel('Firm1 Knowledge');
ylabel('Firm2 Knowledge');
zlabel('Value Func');
fprintf('Time:%f sec',toc);

%% Questions 2&3

[D_0,D_1,D_2] = Ds(p,p,v);
T = zeros(L^2,L^2);

for k=1:30
    if k==1
        for i=1:30
            if i==1
                T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)...
                    +D_1(k,i)*(1-(1-delta)^(k+1))+D_2(k,i)*(1-(1-
```

```
delta)^(i+1));
                T((k-1)*30+i,(k-1)*30+i+1)=D_2(k,i)*(1-delta)^(i+1);
                T((k-1)*30+i,(k)*30+i)=D_1(k,i)*((1-delta)^(k+1));
            elseif i<30
                T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^i)...
                    +D_1(k,i)*(1-(1-delta)^(k+1))*((1-delta)^i)+D_2(k,i)*(1-(1-delta)^(i+1));
                T((k-1)*30+i,(k-1)*30+i+1)=D_2(k,i)*(1-delta)^(i+1);
                T((k-1)*30+i,(k)*30+i)=D_1(k,i)*((1-delta)^(k+1))*(1-delta)^i;
                T((k-1)*30+i,(k-1)*30+i-1)=D_0(k,i)*(1-(1-delta)^i)+D_1(k,i)*(1-(1-delta)^(k+1))*(1-(1-delta)^i);
                T((k-1)*30+i,(k)*30+i-1)=D_1(k,i)*((1-delta)^(k+1))*(1-(1-delta)^i);
            elseif i==30
                T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^i)...
                    +D_1(k,i)*(1-(1-delta)^(k+1))*((1-delta)^i)+D_2(k,i);
                T((k-1)*30+i,(k-1)*30+i-1)=D_0(k,i)*(1-(1-delta)^i)+D_1(k,i)*(1-(1-delta)^(k+1))*(1-(1-delta)^i);
                T((k-1)*30+i,(k)*30+i)=D_1(k,i)*((1-delta)^(k+1))*(1-delta)^i;
                T((k-1)*30+i,(k)*30+i-1)=D_1(k,i)*((1-delta)^(k+1))*(1-(1-delta)^i);
            end
        end
    elseif k<30
        for i=1:30
            if i==1
                T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^k)...
                +D_1(k,i)*(1-(1-delta)^(k+1))+D_2(k,i)*((1-delta)^k)*(1-(1-delta)^(i+1));
                T((k-1)*30+i,(k-1)*30+i+1)=D_2(k,i)*((1-delta)^k)*(1-delta)^(i+1);
                T((k-1)*30+i,(k)*30+i)=D_1(k,i)*((1-delta)^(k+1));
```

```
            T((k-1)*30+i,(k-2)*30+i)=D_0(k,i)*(1-(1-delta)^k)+D_2(k,i)*(1-(1-delta)^k)*(1-(1-delta)^(i+1));
            T((k-1)*30+i,(k-2)*30+i+1)=D_2(k,i)*(1-(1-delta)^k)*((1-delta)^(i+1));

        elseif i<30
            T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^k)*((1-delta)^i)...
              +D_1(k,i)*(1-(1-delta)^(k+1))*((1-delta)^i)+D_2(k,i)*((1-delta)^k)*(1-(1-delta)^(i+1));
            T((k-1)*30+i,(k-1)*30+i+1)=D_2(k,i)*((1-delta)^k)*(1-delta)^(i+1);
            T((k-1)*30+i,(k)*30+i)=D_1(k,i)*((1-delta)^(k+1))*(1-delta)^i;
            T((k-1)*30+i,(k)*30+i-1)=D_1(k,i)*((1-delta)^(k+1))*(1-(1-delta)^i);
            T((k-1)*30+i,(k-1)*30+i-1)=D_0(k,i)*((1-delta)^k)*(1-(1-delta)^i)+D_1(k,i)*(1-(1-delta)^(k+1))*(1-(1-delta)^i);
            T((k-1)*30+i,(k-2)*30+i)=D_0(k,i)*(1-(1-delta)^k)*((1-delta)^i)+D_2(k,i)*(1-(1-delta)^k)*(1-(1-delta)^(i+1));
            T((k-1)*30+i,(k-2)*30+i-1)=D_0(k,i)*(1-(1-delta)^k)*(1-(1-delta)^i);
            T((k-1)*30+i,(k-2)*30+i+1)=D_2(k,i)*(1-(1-delta)^k)*((1-delta)^(i+1));

        elseif i==30
            T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^k)*((1-delta)^i)...
              +D_1(k,i)*(1-(1-delta)^(k+1))*((1-delta)^i)+D_2(k,i)*((1-delta)^k);
            T((k-1)*30+i,(k)*30+i)=D_1(k,i)*((1-delta)^(k+1))*(1-delta)^i;
            T((k-1)*30+i,(k)*30+i-1)=D_1(k,i)*((1-delta)^(k+1))
```

```
                    *(1-(1-delta)^i);
                T((k-1)*30+i,(k-1)*30+i-1)=D_0(k,i)*((1-delta)^k)*(1-(1-delta)^i)+D_1(k,i)*(1-(1-delta)^(k+1))*(1-(1-delta)^i);
                T((k-1)*30+i,(k-2)*30+i)=D_0(k,i)*(1-(1-delta)^k)*((1-delta)^i)+D_2(k,i)*(1-(1-delta)^k);
                T((k-1)*30+i,(k-2)*30+i-1)=D_0(k,i)*(1-(1-delta)^k)*(1-(1-delta)^i);
            end
        end
    elseif k==30
        for i=1:30
            if i==1
                T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^k)...
                +D_1(k,i)+D_2(k,i)*((1-delta)^k)*(1-(1-delta)^(i+1));
                T((k-1)*30+i,(k-1)*30+i+1)=D_2(k,i)*((1-delta)^k)*(1-delta)^(i+1);
                T((k-1)*30+i,(k-2)*30+i)=D_0(k,i)*(1-(1-delta)^k)+D_2(k,i)*(1-(1-delta)^k)*(1-(1-delta)^(i+1));
                T((k-1)*30+i,(k-2)*30+i+1)=D_2(k,i)*(1-(1-delta)^k)*((1-delta)^(i+1));

            elseif i<30
                T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^k)*((1-delta)^i)...
                 +D_1(k,i)*((1-delta)^i)+D_2(k,i)*((1-delta)^k)*(1-(1-delta)^(i+1));
                T((k-1)*30+i,(k-1)*30+i+1)=D_2(k,i)*((1-delta)^k)*(1-delta)^(i+1);
                T((k-1)*30+i,(k-1)*30+i-1)=D_0(k,i)*((1-delta)^k)*(1-(1-delta)^i)+D_1(k,i)*(1-(1-delta)^i);
                T((k-1)*30+i,(k-2)*30+i)=D_0(k,i)*(1-(1-delta)^k)*((1-delta)^i)+D_2(k,i)*(1-(1-delta)^k)*(1-(1-delta)^(i+1));
```

```
                    T((k-1)*30+i,(k-2)*30+i-1)=D_0(k,i)*(1-(1-delta)^k)*(1-(1-delta)^i);
                    T((k-1)*30+i,(k-2)*30+i+1)=D_2(k,i)*(1-(1-delta)^k)*((1-delta)^(i+1));
               elseif i==30
                    T((k-1)*30+i,(k-1)*30+i)=D_0(k,i)*((1-delta)^k)*((1-delta)^i)...
                      +D_1(k,i)*((1-delta)^i)+D_2(k,i)*((1-delta)^k);
                    T((k-1)*30+i,(k-1)*30+i-1)=D_0(k,i)*((1-delta)^k)*(1-(1-delta)^i)+D_1(k,i)*(1-(1-delta)^i);
                    T((k-1)*30+i,(k-2)*30+i)=D_0(k,i)*(1-(1-delta)^k)*((1-delta)^i)+D_2(k,i)*(1-(1-delta)^k);
                    T((k-1)*30+i,(k-2)*30+i-1)=D_0(k,i)*(1-(1-delta)^k)*(1-(1-delta)^i);
               end
          end
     end
end

initial = [1,zeros(1,899)];

figure(3);%10 periods
A1=reshape((initial*(T^10))',30,30);
mesh(A1);
xlabel('Firm1 Knowledge'); ylabel('Firm2 Knowledge'); zlabel('Density');
title('Distribution of states after 10 periods');

figure(4);%20 periods
A2=reshape((initial*(T^20))',30,30);
mesh(A2);
xlabel('Firm1 Knowledge'); ylabel('Firm2 Knowledge'); zlabel('Density');
title('Distribution of states after 20 periods');
```

```
figure(5);%30 periods
A3=reshape((initial*(T^30))',30,30);
mesh(A3);
xlabel('Firm1 Knowledge'); ylabel('Firm2 Knowledge'); zlabel('
    Density');
title('Distribution of states after 30 periods');

figure(6);%Steady State
A4=reshape((initial*(T^900))',30,30);
mesh(A4);
xlabel('Firm1 Knowledge'); ylabel('Firm2 Knowledge'); zlabel('
    Density');
title('Stationary Distribution of states');
```