Homework 7

Kensuke Suzuki

February 15, 2019

Problem 1

I will explain the algorithm. Subscript n of ω_n denotes state of player $n \in 1, 2$ while superscript $i(j)$ of $\omega^{i(j)}$ denotes level of state $i(j) \in \{1, 2, ..., L\}$.

- Get transition matrix $\Pr(\omega'|\omega, q)$ (specified in `setParams.m`).

$$\mathbf{Pr}_q = \begin{bmatrix} \Pr(\omega^1|\omega^1, q) & \Pr(\omega^2|\omega^1, q) & \cdots & \Pr(\omega^L|\omega_1, q) \\ \Pr(\omega^1|\omega^2, q) & \Pr(\omega^2|\omega^2, q) & \cdots & \Pr(\omega^L|\omega_2, q) \\ \vdots & \vdots & \vdots & \cdots \\ \Pr(\omega^1|\omega^L, q) & \Pr(\omega^2|\omega^L, q) & \cdots & \Pr(\omega^L|\omega_L, q) \end{bmatrix}$$

  with representative element $\Pr_q(\omega^i, \omega^j) = \Pr(\omega^j|\omega^i, q)$

- Specify initial guess on $p_1^\ell(\omega_1, \omega_2)$. For $\ell = 0$, I use

$$\mathbf{p}_1^0 = \begin{bmatrix} \frac{c(\omega_1^1)+v}{2} & \frac{c(\omega_1^1)+v}{2} & \dots & \frac{c(\omega_1^1)+v}{2} \\ \frac{c(\omega_1^2)+v}{2} & \frac{c(\omega_1^2)+v}{2} & \dots & \frac{c(\omega_1^2)+v}{2} \\ \vdots & \vdots & \vdots & \dots \\ \frac{c(\omega_1^L)+v}{2} & \frac{c(\omega_1^L)+v}{2} & \dots & \frac{c(\omega_1^L)+v}{2} \end{bmatrix}$$

  with representative ij element is $p^0(\omega_1^i, \omega_2^j) = \frac{c(\omega_1^i)+v}{2}$.

- Specify initial guess $V^\ell(\omega_1, \omega_2)$.

$$\mathbf{V}_1^\ell = \begin{bmatrix} \frac{p^\ell(\omega_1^1,\omega_2^1)-c(\omega_1^1)}{1-\beta} & \frac{p^\ell(\omega_1^1,\omega_2^2)-c(\omega_1^1)}{1-\beta} & \dots & \frac{p^\ell(\omega_1^1,\omega_2^L)-c(\omega_1^1)}{1-\beta} \\ \frac{p^\ell(\omega_1^2,\omega_2^1)-c(\omega_1^2)}{1-\beta} & \frac{p^\ell(\omega_1^2,\omega_2^2)-c(\omega_1^2)}{1-\beta} & \dots & \frac{p^\ell(\omega_1^2,\omega_2^L)-c(\omega_1^2)}{1-\beta} \\ \vdots & \vdots & \vdots & \dots \\ \frac{p^\ell(\omega_1^L,\omega_2^1)-c(\omega_1^L)}{1-\beta} & \frac{p^\ell(\omega_1^L,\omega_2^2)-c(\omega_1^L)}{1-\beta} & \dots & \frac{p^\ell(\omega_1^L,\omega_2^L)-c(\omega_1^L)}{1-\beta} \end{bmatrix}$$

with representative ij element is $V_1^\ell(\omega_1^i, \omega_2^j) = \frac{p^\ell(\omega_1^i, \omega_2^j) - c(\omega_1^i)}{1-\beta}$.

- Get $W_0^\ell(\omega_1, \omega_2)$, $W_1^\ell(\omega_1, \omega_2)$, and $W_2^\ell(\omega_1, \omega_2)$ using function `getW.m`.

$$\mathbf{W}_1^\ell = \mathbf{Pr}_{q=0}\left[\mathbf{V}_1^\ell \mathbf{Pr}_{q=0}^\top\right]$$
$$\mathbf{W}_2^\ell = \mathbf{Pr}_{q=1}\left[\mathbf{V}_1^\ell \mathbf{Pr}_{q=0}^\top\right]$$
$$\mathbf{W}_3^\ell = \mathbf{Pr}_{q=0}\left[\mathbf{V}_1^\ell \mathbf{Pr}_{q=1}^\top\right]$$

- Use built-in solver `fsolve` to solve for first order conditions. I solve function `focp.m` for price vector. Within this function, I use function `D.m` which returns demand for each player for given price matrix. The updated price matrix is denoted by $\mathbf{p}_1^{\ell+1}$.
- Use the $\mathbf{p}_1^{\ell+1}$ to get updated value function $\mathbf{V}_1^{\ell+1}$. Function `getV.m` requires the three inputs: price for player 1, price for player 2, and W. For player 1's price, I use the updated price matrix $\mathbf{p}_1^{\ell+1}$. For player 2's price, I use the initial guess $\left[\mathbf{p}_1^\ell\right]^\top$.
- Use the updated matrices for price and value function, iterate the above steps until policy function and value function are converged.

Figure 1: Value function

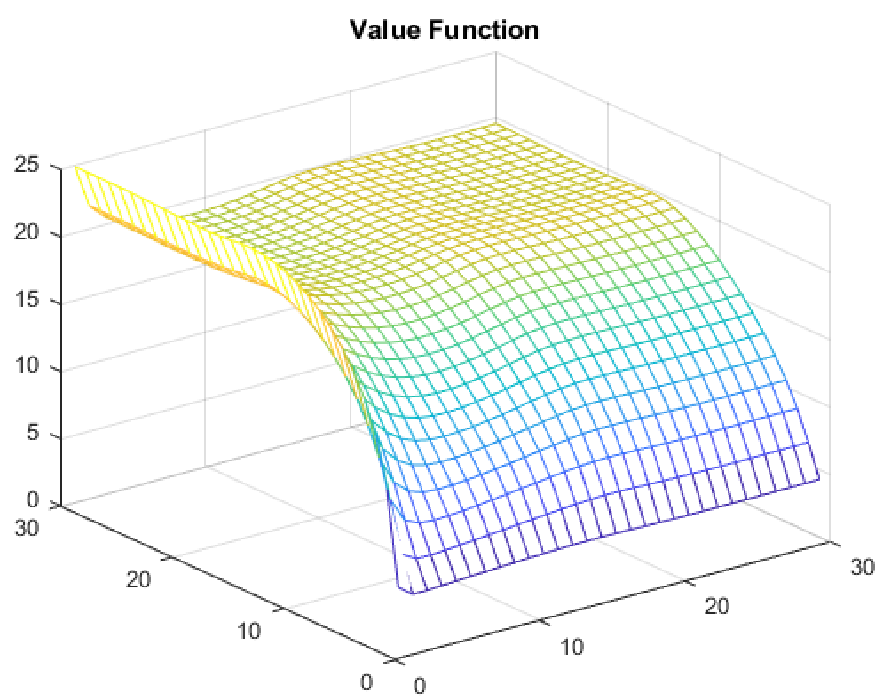
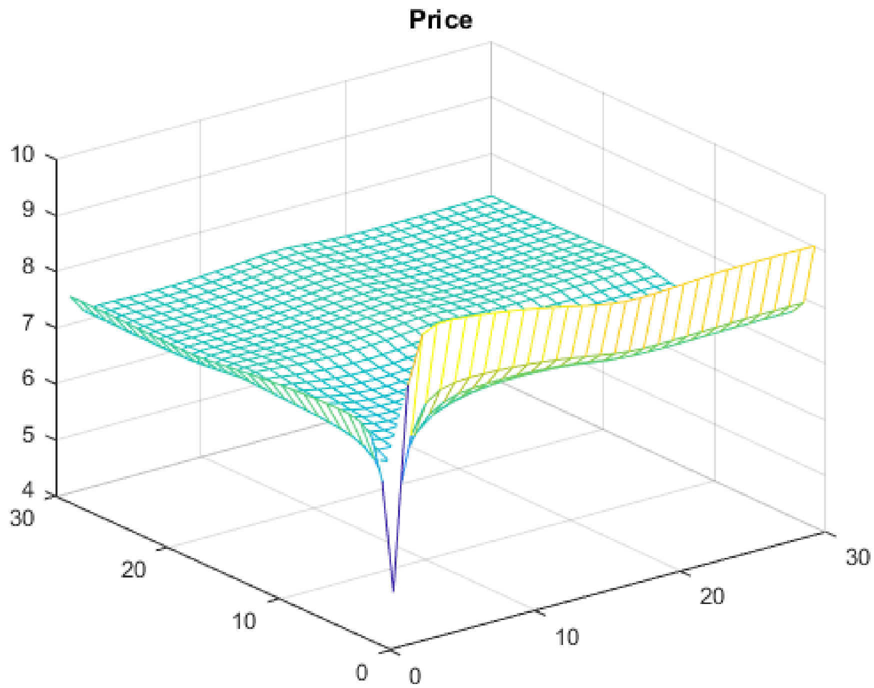


Figure 2: Policy function



Problem 2

In main code, I construct 900×900 transition probability matrix which contains $\Pr(\omega_1', \omega_2' | \omega_1, \omega_2)$:

$$\boldsymbol{\rho} =$$
$$\begin{bmatrix}
\Pr\left(\omega_1^1, \omega_2^1 | \omega_1^1, \omega_2^1\right) & \Pr\left(\omega_1^1, \omega_2^2 | \omega_1^1, \omega_2^1\right) & \cdots & \Pr\left(\omega_1^1, \omega_2^L | \omega_1^1, \omega_2^1\right) & \Pr\left(\omega_1^2, \omega_2^1 | \omega_1^1, \omega_2^1\right) & \cdots & \Pr\left(\omega_1^L, \omega_2^L | \omega_1^1, \omega_2^1\right) \\
\Pr\left(\omega_1^1, \omega_2^1 | \omega_1^1, \omega_2^2\right) & \Pr\left(\omega_1^1, \omega_2^2 | \omega_1^1, \omega_2^2\right) & \cdots & \Pr\left(\omega_1^1, \omega_2^L | \omega_1^1, \omega_2^2\right) & \Pr\left(\omega_1^2, \omega_2^1 | \omega_1^1, \omega_2^2\right) & \cdots & \Pr\left(\omega_1^L, \omega_2^L | \omega_1^1, \omega_2^2\right) \\
\cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\Pr\left(\omega_1^1, \omega_2^1 | \omega_1^1, \omega_2^L\right) & \Pr\left(\omega_1^1, \omega_2^2 | \omega_1^1, \omega_2^L\right) & \cdots & \Pr\left(\omega_1^1, \omega_2^L | \omega_1^1, \omega_2^L\right) & \Pr\left(\omega_1^2, \omega_2^1 | \omega_1^1, \omega_2^L\right) & \cdots & \Pr\left(\omega_1^L, \omega_2^L | \omega_1^1, \omega_2^L\right) \\
\Pr\left(\omega_1^1, \omega_2^1 | \omega_2^2, \omega_2^1\right) & \Pr\left(\omega_1^1, \omega_2^2 | \omega_2^2, \omega_2^1\right) & \cdots & \Pr\left(\omega_1^1, \omega_2^L | \omega_2^2, \omega_2^1\right) & \Pr\left(\omega_1^2, \omega_2^1 | \omega_2^2, \omega_2^1\right) & \cdots & \Pr\left(\omega_1^L, \omega_2^L | \omega_2^2, \omega_2^1\right) \\
\cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\Pr\left(\omega_1^1, \omega_2^1 | \omega_2^L, \omega_2^L\right) & \Pr\left(\omega_1^1, \omega_2^2 | \omega_2^L, \omega_2^L\right) & \cdots & \Pr\left(\omega_1^1, \omega_2^L | \omega_2^L, \omega_2^L\right) & \Pr\left(\omega_1^2, \omega_2^1 | \omega_2^L, \omega_2^L\right) & \cdots & \Pr\left(\omega_1^L, \omega_2^L | \omega_2^L, \omega_2^L\right)
\end{bmatrix}$$

In order to construct $\Pr(\omega_1', \omega_2' | \omega_1, \omega_2)$, I use the following relationship

$$\begin{aligned}
\Pr(\omega_1', \omega_2' | \omega_1, \omega_2) &= \Pr(\omega_1', \omega_2' | \omega_1, \omega_2, q_1, q_2) \Pr(q_1, q_2 | \omega_1, \omega_2) \\
&= \Pr(\omega_1' | \omega_1, q_1) \Pr(\omega_2' | \omega_2, q_2) \Pr(q_1, q_2 | \omega_1, \omega_2)
\end{aligned}$$

and

$$\Pr(q_1, q_2|\omega_1, \omega_2) = \begin{cases} D_0 & \text{if} \quad q_1 = q_2 = 0 \\ D_1 & \text{if} \quad q_1 = 1, \; q_2 = 0 \\ D_2 & \text{if} \quad q_1 = 0, \; q_2 = 1 \end{cases}$$

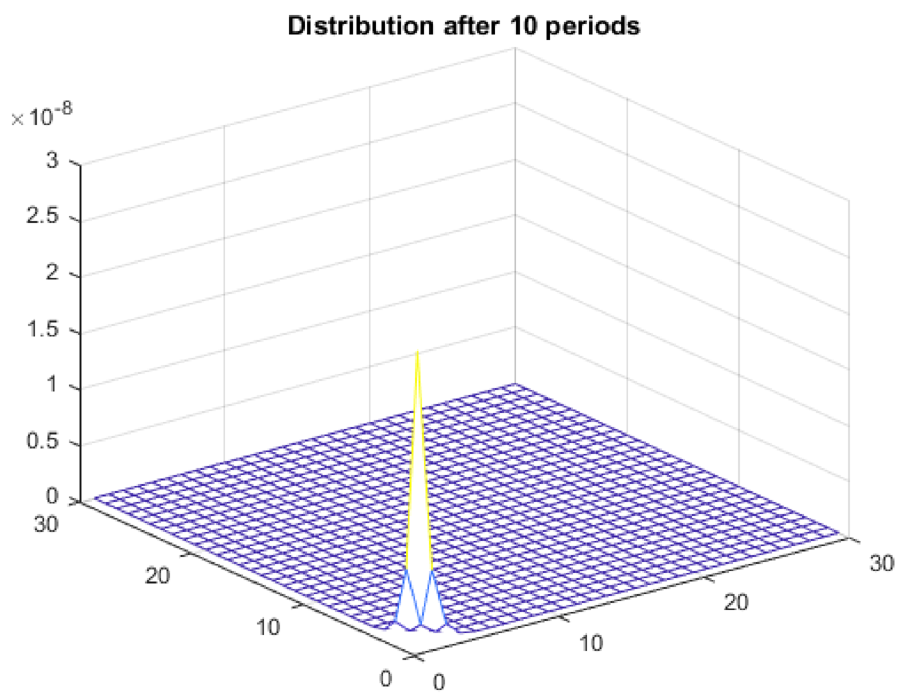
Since I know $\Pr(\omega_1'|\omega_1, q_1)$, $\Pr(\omega_2'|\omega_2, q_2)$, and $\Pr(q_1, q_2|\omega_1, \omega_2)$, we can construct **Trans** using the price matrix obtained in problem 1.
Let the state vector be $\mathbf{s} \in \mathbb{R}^{900}$ where each element corresponds to unique pair of (ω_1, ω_2). For example, the first element is (ω_1^1, ω_2^1), the second element is (ω_1^1, ω_2^L), and the last element is (ω_1^L, ω_2^L). Since the initial state is (1,1), let $\mathbf{s}^1 = [1, 0, \cdots, 0]^\top$. The probability distribution in period 2 is

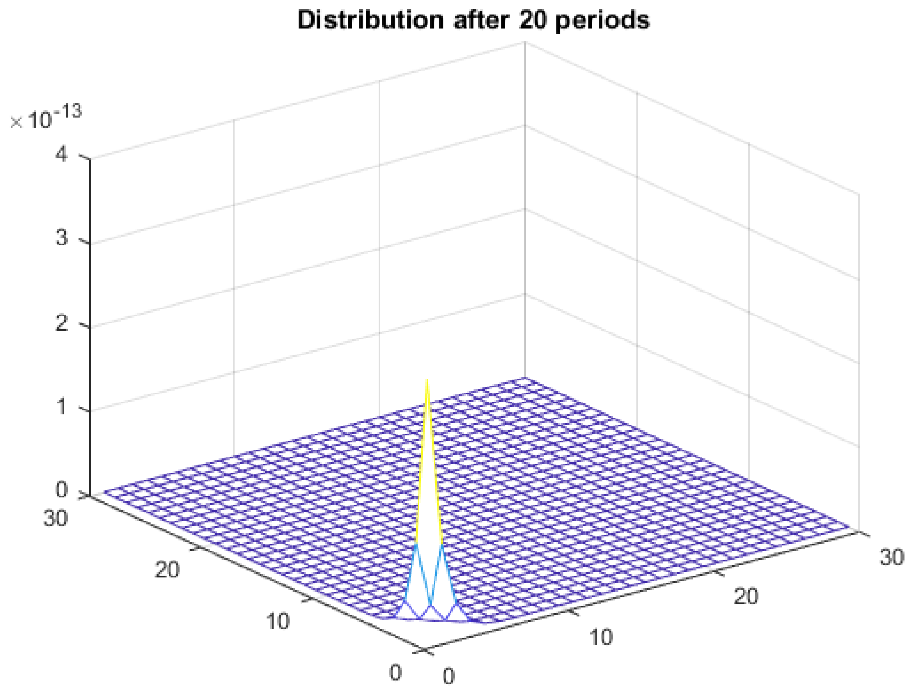
$$\mathbf{s}^2 = \boldsymbol{\rho}\mathbf{s}^1$$

We can repeat this to get the probability distribution over state in period t as

$$\mathbf{s}^t = \boldsymbol{\rho}\mathbf{s}^{t-1}$$

Following images demonstrate the probability distribution after 10, 20, and 30 periods.



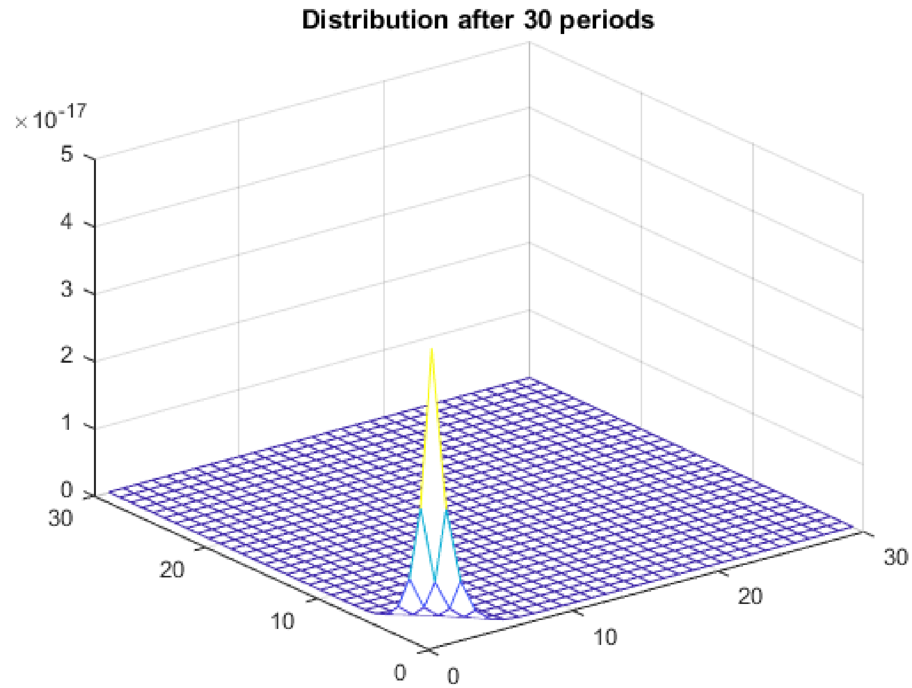


Problem 3

Stationary distribution is obtained by iterating

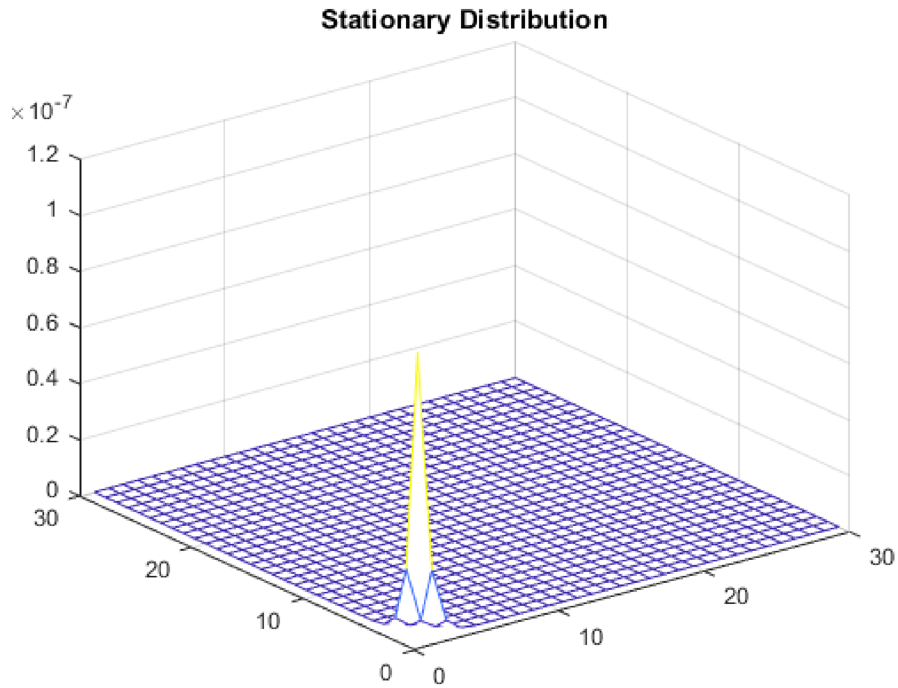
$$\mathbf{s}^t = \boldsymbol{\rho}\mathbf{s}^{t-1}$$

for $t = 2, 3, ...$ until $\mathbf{s}^t = \mathbf{s}^{t+1}$



Matlab Main Code

```
clear;
tic;
setupParams;

%% Problem 1

% Initial guess
cmat = kron(c, ones(1,L));
p1 = (repmat(c,1,L) + v)./2;
V1 = (p1 - cmat)./(1-beta);
V1 = ones(L,L);

diff = 1;
iter=1;

while diff > 1e-5 && iter < 100000;

    W = getW(V1);

    psol = @(p) focp(p,W);
    p1_new = fsolve(psol, p1);

    V1_new = getV(p1,p1_new,W);

    diff_p = max( abs(((p1_new - p1)./(1+abs(p1_new)))) );
```




```
    diff_V = max( abs(((V1_new - V1)./(1+abs(V1_new)))) );
    diff = max([diff_p,diff_V])


    V1 = lambda .* V1_new + (1-lambda).* V1;
    p1 = lambda .* p1_new + (1-lambda).* p1;
    iter = iter +1

end

figure(1);
mesh(V1);
title('Value Function');

figure(2);
mesh(p1);
title('Price');


%% Problem 2
% transition matrix
% Raw: (omega1, omega2)
% Column (omega1', omega2')
Trans = zeros(L*L,L*L);

for i = 1:L*L
    for j = 1:L*L
```

```
        if rem(i,30) == 0
            i1 = fix(i/30);
        else
            i1 = fix(i/30) +1 ;
        end
        i2 = rem(i,30);
        if i1 == 31
            i1 = 30;
        end
        if i2 == 0
            i2 = 30;
        end

        if rem(j,30) == 0
            j1 = fix(j/30);
        else
            j1 = fix(j/30) +1 ;
        end

        j2 = rem(j,30);
        if j1 == 31
            j1 = 30;
        end
        if j2 == 0
            j2 = 30;
        end
        [i,j];
        ipair = [i1,i2]; % today's state for player 1 and 2
        jpair = [j1,j2]; % future state


        Trans(i,j)  = Pr(i1,j1,1)*Pr(i2,j2,1)*(1 - D(p1(i1,i2),p1(i2,i1))-
           D(p1(i2,i1), p1(i1,i2)) ) ...
                     + Pr(i1,j1,2)*Pr(i2,j2,1)*(D(p1(i1,i2),p1(i2,i1)) ) ...
                     + Pr(i1,j1,1)*Pr(i2,j2,2)*(D(p1(i2,i1), p1(i1,i2)) ) ;

    end
end

clear state state_new

% initial state
state_int = zeros(L*L,1);
state_int(1,1) = 1;

% 10 period
state = state_int;
for t = 2:10
    state_new = Trans * state;
    state = state_new;
```

```
end

Dstrbn10 = zeros(L,L);
for i = 1:L
    Dstrbn10(i,:) = (state_new((i-1)*L+1:i*L))';
end

% 20 period
state = state_int;
for t = 2:20
    state_new = Trans * state;
    state = state_new;
end

Dstrbn20 = zeros(L,L);
for i = 1:L
    Dstrbn20(i,:) = (state_new((i-1)*L+1:i*L))';
end

% 30 period
state = state_int;
for t = 2:30
    state_new = Trans * state;
    state = state_new;
end

Dstrbn30 = zeros(L,L);
for i = 1:L
    Dstrbn30(i,:) = (state_new((i-1)*L+1:i*L))';
end

figure(3);
mesh(Dstrbn10);
title('Distribution after 10 periods');

figure(4);
mesh(Dstrbn20);
title('Distribution after 20 periods');

figure(5);
mesh(Dstrbn30);
title('Distribution after 30 periods');

%% Problem 3
% stationary distribution

state = state_int;
diff = 1;
while diff > 1e-6
    state_new = Trans * state;
```

```
    diff = max(abs(state - state_new))
    state = state_new;
end

StDstrbn = zeros(L,L);
for i = 1:L
    StDstrbn(i,:) = (state_new((i-1)*L+1:i*L))';
end

figure(6);
mesh(StDstrbn);
title('Stationary Distribution');
```