Macroeconomics III – Homework 6

Hopenhayn Firm Dynamics

1. Tauchen Approximation of Productivity

Productivity follows an AR(1):

$$s_{t+1} = \rho s_t + \varepsilon_{t+1}, \qquad \varepsilon \sim N(0, \sigma^2).$$

We apply the Tauchen (1986) method with parameters

$$\rho = 0.9, \qquad \sigma = 0.1, \qquad N = 21.$$

The unconditional variance is:

$$\sigma_s^2 = \frac{\sigma^2}{1 - \rho^2}.$$

Figure 1 plots the resulting discrete grid for log productivity.

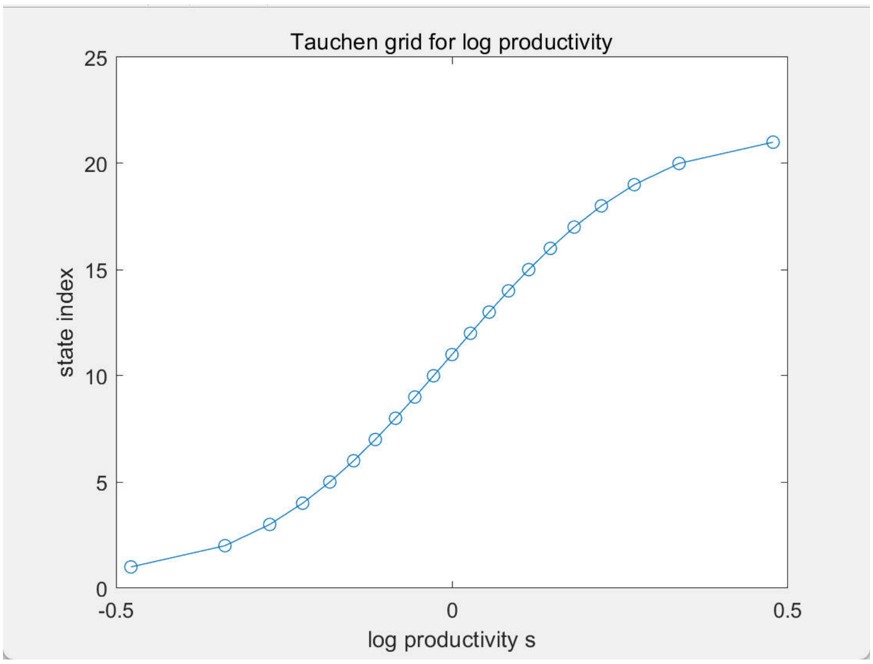

Figure 1: Tauchen grid for log productivity

2. Free Entry Condition and $W_e(w)$

An entrant draws initial productivity from $N(\mu_\nu, \sigma_\nu)$ with

$$\mu_\nu = -0.2, \qquad \sigma_\nu = 0.1.$$

Given wage w, the value of entry is:

$$W_e(w) = \mathbb{E}_\nu[V(s_0)] - c_e, \qquad c_e = 0.04.$$

Using the MATLAB implementation, we evaluate $W_e(w)$ over $w \in [0.5, 3]$. The result is shown in Figure 2.

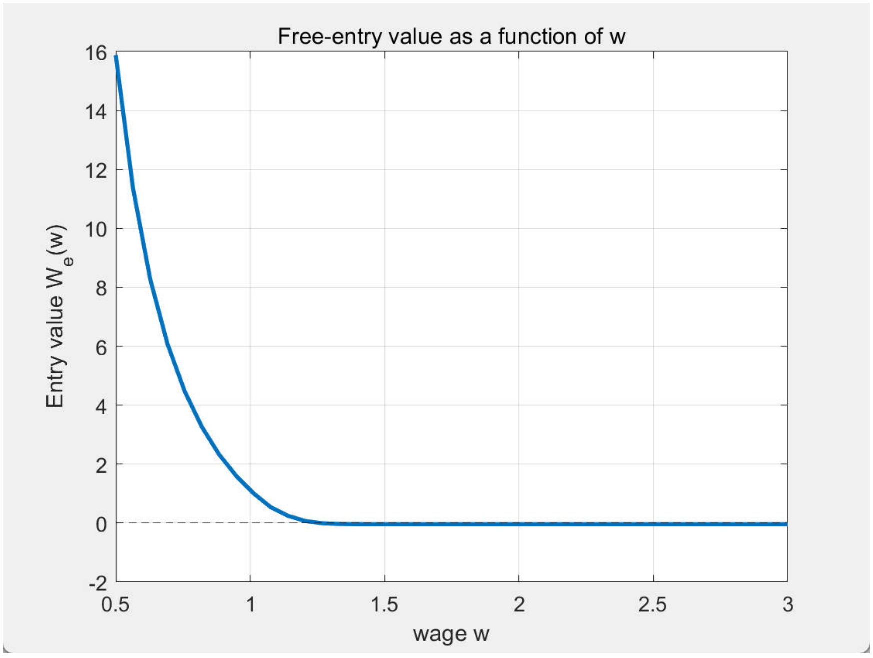

Figure 2: Free-entry value $W_e(w)$

The function is strictly decreasing. The root gives the equilibrium wage.

$$\boxed{w^* = 1.260511}$$

3. Firm Problem and Exit Rule at w^*

The incumbent firm chooses labor

$$n(s) = \left(\frac{\theta A e^s}{w}\right)^{\frac{1}{1-\theta}}, \quad \theta = 0.7, \; A = 1,$$

with profit

$$\pi(s) = Ae^s n(s)^\theta - wn(s) - c_f, \qquad c_f = 0.1.$$

The Bellman equation is

$$V(s) = \max\left\{\pi(s) + \beta \sum_{s'} P(s,s')V(s'),\, 0\right\}, \quad \beta = 0.96.$$

Figure 3 shows the value function and exit states.

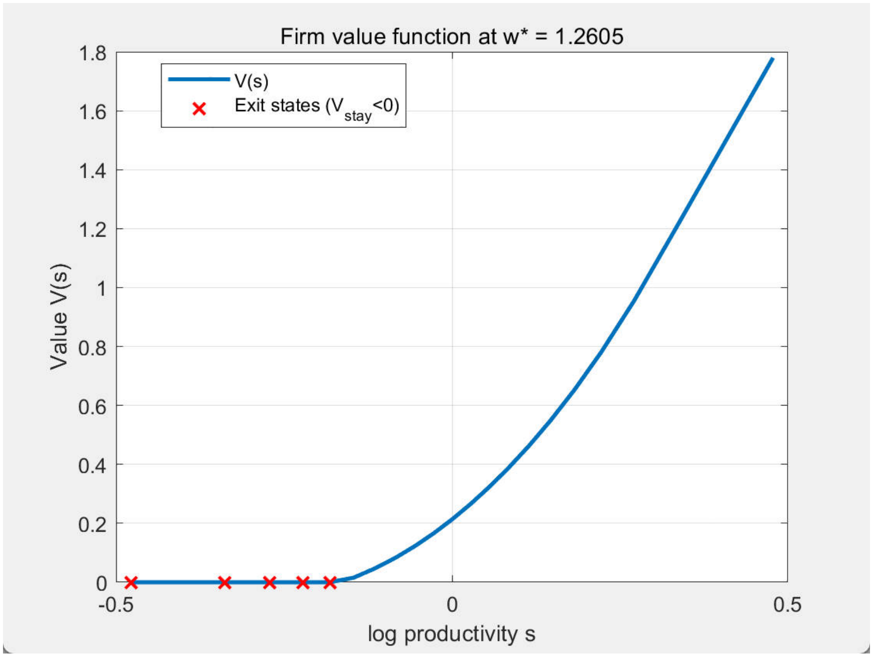

Figure 3: Value function $V(s)$ and exit states at $w^* = 1.2605$

Low-productivity firms optimally exit, consistent with Hopenhayn (1992).

4. Stationary Distribution and Aggregates for $M = 1$

Let Q be the transition matrix truncated by exit:

$$Q_{ij} = \begin{cases} P_{ij}, & \text{if firm continues in state } i, \\ 0, & \text{if firm exits.} \end{cases}$$

The stationary measure of firms is:

$$\hat{\mu} = (I - Q')^{-1}\eta,$$

where η is the distribution of entrants.

The model produces:

$$\sum_s \hat{\mu}(s) = 7.8138, \qquad \Pi(\hat{\mu}) = 0.115444, \qquad L_d(\hat{\mu}) = 1.660118.$$

5. General Equilibrium with Labor Market Clearing

Household preferences are:

$$u(c,n) = \ln c + \xi \ln(1-n), \qquad \xi = 1.$$

Budget constraint (per capita):

$$c = wn + M\Pi(\hat{\mu}).$$

Labor supply FOC gives:

$$n^s(M) = \frac{w - \xi M \Pi(\hat{\mu})}{w(1+\xi)}, \qquad 0 \le n^s \le 1.$$

Labor demand from firms:

$$L_d(M) = M L_d(\hat{\mu}).$$

Equilibrium requires:

$$L_d(M^*) = n^s(M^*).$$

Solving numerically:

$$\boxed{M^* = 0.293099}$$

$$L_d(M^*) = 0.486578, \qquad n^s(M^*) = 0.486578.$$

Thus the goods, labor, and entry markets all clear.

6. MATLAB Code

```
%% Macroeconomics III - Homework 6
% Hopenhayn firm dynamics
clear; clc; close all;


```

```
function We = firm_entry_value(w, grid_s, P, params)
% Returns entry value We(w) for given wage w

[We, ~, ~, ~, ~, ~] = solve_firm(w, grid_s, P, params);

end

function [We, V, exit_policy, n_star, pi, entry_dist] = ...
    solve_firm(w, grid_s, P, params)
% Solve incumbent firm's problem for given wage w
% and compute entry value We(w).

A        = params.A;
theta    = params.theta;
beta     = params.beta;
cf       = params.cf;
ce       = params.ce;
mu_nu    = params.mu_nu;
sigma_nu = params.sigma_nu;

N = length(grid_s);

% -------- Static profit maximization (labor choice) --------
z = exp(grid_s(:));  % productivity levels (column)

% optimal labor choice: n* = (theta * A * exp(s) / w)^{1/(1-theta
    )}
n_star = (theta * A .* z / w) .^ (1 / (1 - theta));

y  = A .* z .* (n_star .^ theta);       % output
pi = y - w .* n_star - cf;              % per-period profit (
    before exit decision)

% -------- Dynamic programming for incumbent --------
V = zeros(N,1);          % initial guess
tol   = 1e-8;
maxit = 10000;

for it = 1:maxit
    cont    = beta * (P * V);       % continuation value E[V' | s]
    V_stay  = pi + cont;           % value if stay this period
```

```
    V_new    = max(V_stay, 0);      % exit value normalized to 0

    diff = max(abs(V_new - V));
    V    = V_new;

    if diff < tol
        break;
    end
end

% exit_policy(i) = 1 if firm exits at state i
exit_policy = V_stay < 0;

% -------- Entry value We(w) --------
% initial log-productivity s_0 ~ N(mu_nu, sigma_nu)
entry_dist = normpdf(grid_s, mu_nu, sigma_nu);
entry_dist = entry_dist(:) / sum(entry_dist);  % normalize to 1

We = entry_dist' * V - ce;

end

function [grid,P] = Tauchen(rho,N,sigma2, mue,tol)
%TAUCHEN Generates a discrete approximation to an AR-1 Process

if nargin<5
    tol = 0;
end

sigmaAR = sqrt(sigma2);


grid_probs=linspace(0,1,N+1);    % generate equi-likely bins
bounds=norminv(grid_probs,mue,sigmaAR);      % corresponding bin
    bounds

% Calculate grid - centers
grid = sigmaAR*N*( normpdf((bounds(1:end-1)-mue)/sigmaAR) -
    normpdf((bounds(2:end)-mue)/sigmaAR) )+mue;

% replace (-)Inf bounds, by finite numbers
```

```
bounds(1)=bounds(2)-99999999999;
bounds(end)=bounds(end-1)+99999999999;

sigma_e=sqrt((1-rho^2)*sigma2);% Calculate short run std
P=zeros(N,N); % Initialize Transition Probability Matrix

for i=1:N %state
    for j=1:N %transition to
%          pi = normcdf(bounds(j+1),rho*grid_w+(1-rho)*mue,sigma_e)-normcdf(bounds(j),rho*grid_w+(1-rho)*mue,sigma_e);
%          P(i,j)= grid_wp'*pi;
        P(i,j)=normcdf((bounds(j+1)-rho*grid(i)-(1-rho)*mue)/sigma_e)-normcdf((bounds(j)-rho*grid(i)-(1-rho)*mue)/sigma_e);
    end
end

if tol>0                            % If sparse matrix is asked for:
    P(P<tol) = 0;                   % Set entries below tolerance to 0.
end


% Make sure P is a Probability Matrix
P=P./sum(P,2);

if tol>0                            % Make output sparse if asked for.
    P = sparse(P);
end
end

%% ----------------- Parameters -----------------
A       = 1;
theta   = 0.7;
rho     = 0.9;
sigma   = 0.1;            % innovation std. in s_{t+1} = rho s_t + sigma * eps
beta    = 0.96;
cf      = 0.1;           % fixed cost
ce      = 0.04;          % entry cost
```

```
xi      = 1;            % utility parameter in ln c + xi ln(1-n)

mu_nu    = -0.2;        % mean of initial draw distribution
sigma_nu = 0.1;         % std of initial draw distribution

N = 21;                 % number of grid points for s

%% ----------------- (a) Tauchen approximation -----------------
% AR(1): s_{t+1} = rho s_t + eps_{t+1}, eps ~ N(0, sigma^2)
% Tauchen.m (class version) uses unconditional variance sigma2 of s

sigma2 = sigma^2 / (1 - rho^2);  % unconditional variance of s
mue    = 0;                      % mean of s (process centered at 0)

[grid_s, P] = Tauchen(rho, N, sigma2, mue, 0);

% Simple illustration: grid of log-productivity points
figure;
plot(grid_s, 1:N, 'o-');
xlabel('log productivity s');
ylabel('state index');
title('Tauchen grid for log productivity');

%% Pack parameters in a struct for convenience
params.A        = A;
params.theta    = theta;
params.beta     = beta;
params.cf       = cf;
params.ce       = ce;
params.mu_nu    = mu_nu;
params.sigma_nu = sigma_nu;

%% ----------------- (b) Entry value W_e(w) as a function of w -----------------
w_grid = linspace(0.5, 3, 40);
We_grid = zeros(size(w_grid));

for i = 1:length(w_grid)
    w = w_grid(i);
```

```
    We_grid(i) = firm_entry_value(w, grid_s, P, params);
end

figure;
plot(w_grid, We_grid, 'LineWidth', 2);
hold on; yline(0,'k--');
xlabel('wage w');
ylabel('Entry value W_e(w)');
title('Free-entry value as a function of w');
grid on;

%% ----------------- (c) Find equilibrium wage w* from free-entry condition W_e(w*) = 0 ------------------
% Use fzero with a wrapper function

w_lower = 0.5;
w_upper = 3.0;

We_fun = @(w) firm_entry_value(w, grid_s, P, params);  % only returns scalar We
w_star = fzero(We_fun, [w_lower, w_upper]);

fprintf('Equilibrium wage w* (free entry): %.6f\n', w_star);

% Solve firm problem at w*
[We_star, V_star, exit_policy, n_star, pi_star, entry_dist] = ...
    solve_firm(w_star, grid_s, P, params);

% Plot firm value function at w*
figure;
plot(grid_s, V_star, 'LineWidth', 2);
xlabel('log productivity s');
ylabel('Value V(s)');
title(sprintf('Firm value function at w* = %.4f', w_star));
grid on;

% Exit rule: where continuing value is negative, the firm exits (V=0)
stay_states = find(exit_policy == 0);   % 0 = stay, 1 = exit
exit_states = find(exit_policy == 1);   % 1 = exit

```

```matlab
hold on;
plot(grid_s(exit_states), V_star(exit_states), 'rx', 'MarkerSize', 8, 'LineWidth', 1.5);
legend('V(s)', 'Exit states (V_{stay}<0)', 'Location', 'Best');

%% ----------------- (d) Stationary measure \hat{\mu}, profits \Pi, labor demand L^d_hat for M=1 -----------------
% exit_policy(i) = 1 if exit, 0 if stay
stay = 1 - exit_policy;              % 1 if stay, 0 if exit
stay = stay(:);                      % make sure it is column

% Transition matrix of continuing firms: Q(i,j) = P(i,j) if stay(i)=1, 0 otherwise
Q = P .* stay;                       % element-wise multiply rows

I_N = eye(N);

% Entry distribution over states at time of entry (discretized N(mu_nu, sigma_nu))
eta = entry_dist(:);                 % already normalized in solve_firm

% Stationary measure (column vector) satisfying: mu = eta + Q' * mu  when M=1
mu_hat = (I_N - Q') \ eta;

% Profits and labor demand at M=1
Pi_hat = pi_star(:)' * mu_hat;     % total profits
Ld_hat = n_star(:)' * mu_hat;      % total labor demand

fprintf('For entry mass M = 1:\n');
fprintf('  Stationary mass of firms sum(mu_hat): %.4f\n', sum(mu_hat));
fprintf('  Total profits Pi_hat: %.6f\n', Pi_hat);
fprintf('  Labor demand Ld_hat: %.6f\n', Ld_hat);

%% ----------------- (e) Equilibrium entry mass M* from labor-market clearing -----------------
% Labor demand with entry mass M: L_d(M) = M * Ld_hat
% Profits with entry mass M:      Pi(M) = M * Pi_hat
% Representative household: u(c,n) = ln c + xi ln(1-n),
```

```
% budget c = w*n + Pi(M)
% => FOC gives labor supply:
%     n^s(M) = (w - xi*Pi(M)) / (w*(1+xi))

labor_supply = @(M) max(0, min(1, (w_star - xi * (M*Pi_hat)) / (
    w_star * (1 + xi))));
excess_labor = @(M) M * Ld_hat - labor_supply(M);

% Find M* such that L_d(M*) = n^s(M*)
M_lower = 1e-4;
M_upper = 10;

M_star = fzero(excess_labor, [M_lower, M_upper]);

L_d_star = M_star * Ld_hat;
L_s_star = labor_supply(M_star);

fprintf('\nEquilibrium entry mass M* (labor market clearing): %.6
    f\n', M_star);
fprintf('  Labor demand at M*: %.6f\n', L_d_star);
fprintf('  Labor supply at M*: %.6f\n', L_s_star);
```

Conclusion

This homework implements the full Hopenhayn (1992) competitive industry equilibrium with entry, exit, and heterogeneous productivity. The results conform to theory:

- High wages discourage entry ($W_e(w)$ decreasing). - Low-productivity firms exit in equilibrium. - Stationary firm measure is finite even with free entry. - General equilibrium selects a unique wage–entry pair (w^*, M^*).