ECON282B - Problem Set #1

Christopher Saw, Ekaterina Gurkova

January 18, 2022

Question 1

In the symmetric steady state, an entrant with productivity z drawn from cdf $G(z)$ earns revenues:

$$R(z) = \begin{cases} \Pi_D \exp(z) + \Pi_D \tau^{1-\rho} \exp(z), & \text{if } z \geq z_x \\ \Pi_D \exp(z), & \text{otherwise} \end{cases}$$

where $\Pi_D = \frac{P^\rho Y}{\rho^\rho (\rho-1)^{1-\rho}}$ is a domestic demand index.

Let $V_x(z)$ denote the value function of a firm with productivity z that starts the period as an exporter (and has already paid the sunk cost f_{XS} to export):

$$\begin{aligned} V_x(z) = (1+\tau^{1-\rho})\Pi_D \exp(z) - f - f_X + \beta(1-\delta)\Big[&\bar{q} V_x(z + \triangle z) \\ &+ (1-\bar{q})\Big((1-p_n)V_x(z-\triangle z) + p_n V_n(z - \triangle z)\Big)\Big] \end{aligned} \tag{1}$$

where p_n denotes the probability that an exporter draws a sufficiently low $z - \triangle z$ to become a non-exporter:

$$p_n = Pr(z - \triangle z < z_n | z \geq z_n) = \frac{Pr(z_n \leq z < z_n + \triangle z)}{Pr(z \geq z_n)} = \frac{G(z_n + \triangle z) - G(z_n)}{1 - G(z_n)} \tag{2}$$

Let $V_n(z)$ denote the value function of a firm with productivity z that starts the period as a non-exporter (and incurs the sunk cost f_{XS} if it exports):

$$\begin{aligned} V_n(z) = \Pi_D \exp(z) - f + \beta(1-\delta)\Big[&\bar{q}\Big(p_x(V_x(z+\triangle z) - f_{XS}) + (1-p_x)V_n(z+\triangle z)\Big) \\ &+ (1-\bar{q})V_n(z-\triangle z)\Big] \end{aligned} \tag{3}$$

where p_x denotes the probability that a non-exporter draws a sufficiently high $z + \triangle z$ to become an exporter:

$$p_x = Pr(z + \triangle z \geq z_x | z < z_x) = \frac{Pr(z_x - \triangle z \leq z < z_x)}{Pr(z < z_x)} = \frac{G(z_x) - G(z_x - \triangle z)}{G(z_x)} \tag{4}$$

Question 2

An exporter becomes a non-exporter if $\Pi_D \tau^{1-\rho} \exp(z) - f_X \leq 0$ which implies a unique productivity cutoff z_n such that

$$\Pi_D \tau^{1-\rho} \exp(z_n) = f_X \tag{5}$$

and a non-exporter becomes an exporter if $\Pi_D \tau^{1-\rho} \exp(z) - f_X - f_{XS} \geq 0$ which implies a unique productivity cutoff z_x such that

$$\Pi_D \tau^{1-\rho} \exp(z_x) = f_X + f_{XS} \tag{6}$$

Since the LHS is strictly increasing in z, the equations (5) and (6) above show that $z_n < z_x$ if $f_{XS} > 0$ and $z_n = z_x$ if $f_{XS} = 0$. The intuition for $z_n < z_x$ is straightforward. Since the non-exporting firm has to pay an additional sunk cost f_{XS} to become an exporter, a non-exporting firm that switches to exporting must be more productive than another non-exporting firm that does not switch, and more productive than a previously exporting firm that has now become non-exporting.

Question 3

Given a mass of entrants M_e, productivity distribution with pdf $g(z)$, the evolution of firms with productivity index z that are exporters/non-exporters is given by:

$$M_x(z') = \begin{cases} M_e g(z') + (1-\delta)[\bar{q} M_x(z' - \triangle z) + (1-q) M_x(z' + \triangle z)] & \text{if } z' \geq \underline{z}, z' > z_n \\ \\ M_e g(z') + (1-\delta)[\bar{q}(M_x(z' - \triangle z) + M_n(z' - \triangle z)) \\ +(1-q)(M_x(z' + \triangle z)] & \text{if } z' = z_x \\ \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$M_n(z') = \begin{cases} M_e g(z') + (1-\delta)[\bar{q} M_n(z' - \triangle z) + (1-q) M_n(z' + \triangle z)] & \text{if } z' \geq \underline{z}, z' < z_x, \\ & z' \neq z_n \\ M_e g(z') + (1-\delta)[\bar{q} M_n(z' - \triangle z) + \\ (1-q)(M_n(z' + \triangle z) + M_x(z' + \triangle z)] & \text{if } z' = z_n \\ \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Question 4

1. For a given Π_D, use (5) and (6) to calculate the cutoffs $z_n(\Pi_D)$ and $z_x(\Pi_D)$:

$$\begin{aligned} z_n &= \log(f_X) - \log(\Pi_D) - (1-\rho)\log(\tau) \\ z_x &= \log(f_X + f_{XS}) - \log(\Pi_D) - (1-\rho)\log(\tau) \end{aligned}$$

2. Next, use (2) and (4) to calculate the probabilities $p_n(\Pi_D)$ and $p_x(\Pi_D)$.

3. Search over Π_D and solve (1) and (3) and the exit cutoff $\underline{z}$:

$$V_n(\underline{z}) = 0$$

4. Check if Π_D solves the free entry condition (go back to Step 1 if it does not):

$$f_E = \beta\left[\int_0^{z_x} V_n(z)dG(z) + \int_{z_x}^{\bar{z}} V_x(z)dG(z)\right]$$

5. Use (7) and (8) to solve for the steady-state $M_x(z)$ and $M_n(z)$.

6. Use Π_D, M_x and M_n from Steps 3 and 4 with (9) and (10); M_e is the solution to the labor market clearing condition:

$$L_P + f_E M_e + \int_{\underline{z}}^{z_x} f M_n(z)dz + \int_{z_x}^{\bar{z}} (f + f_X)M_x(z)dz + \bar{q} f_{XS} M_n(z_x - \triangle z) = 1 \tag{9}$$

7. And total production labor is given by:

$$L_P = \int_{\underline{z}}^{z_x} l_n(z)M_n(z)dz + \int_{z_x}^{\bar{z}} l_x(z)M_x(z)dz \tag{10}$$

where:

$$\begin{aligned} l_n(z) &= (\rho - 1)\Pi_D \exp(z) \\ l_x(z) &= (\rho - 1)\Pi_D[1 + \tau^{(1-\rho)}]\exp(z) \end{aligned}$$

8. Aggregate output Y is then:

$$Y = L_P\Big(\int_{\underline{z}}^{z_x} \exp(z)M_n(z)dz + \int_{z_x}^{\bar{z}} [1 + \tau^{(1-\rho)}]\exp(z)M_x(z)dz\Big)^{\frac{1}{\rho-1}}$$

Question 5

We assume that z follows a Pareto distribution: $G(z) = 1 - (\frac{z_{min}}{z})^{\theta}$ and use the following values for the model parameters, where the top panel follows Burstein & Melitz:

Parameter	Value
β	$\frac{1}{1.05}$
δ	0.005
ρ	5
τ	0.231
$\triangle z$	0.25
f_E	1
f	0.1
f_X	1.4
θ	4
$\bar{q}$	0.5
f_{XS}	1

Table 1: Model Parameters

Question 6

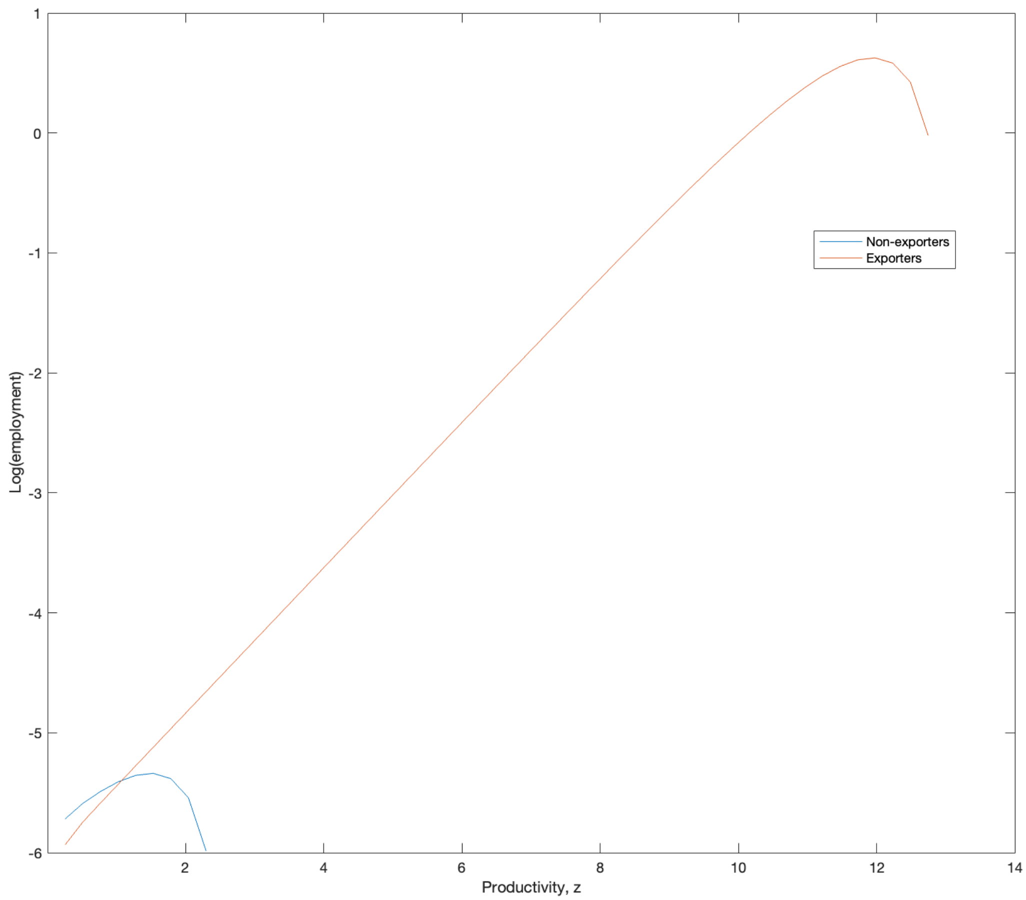

Figure 1: Log(employment)

Question 6 (continued)

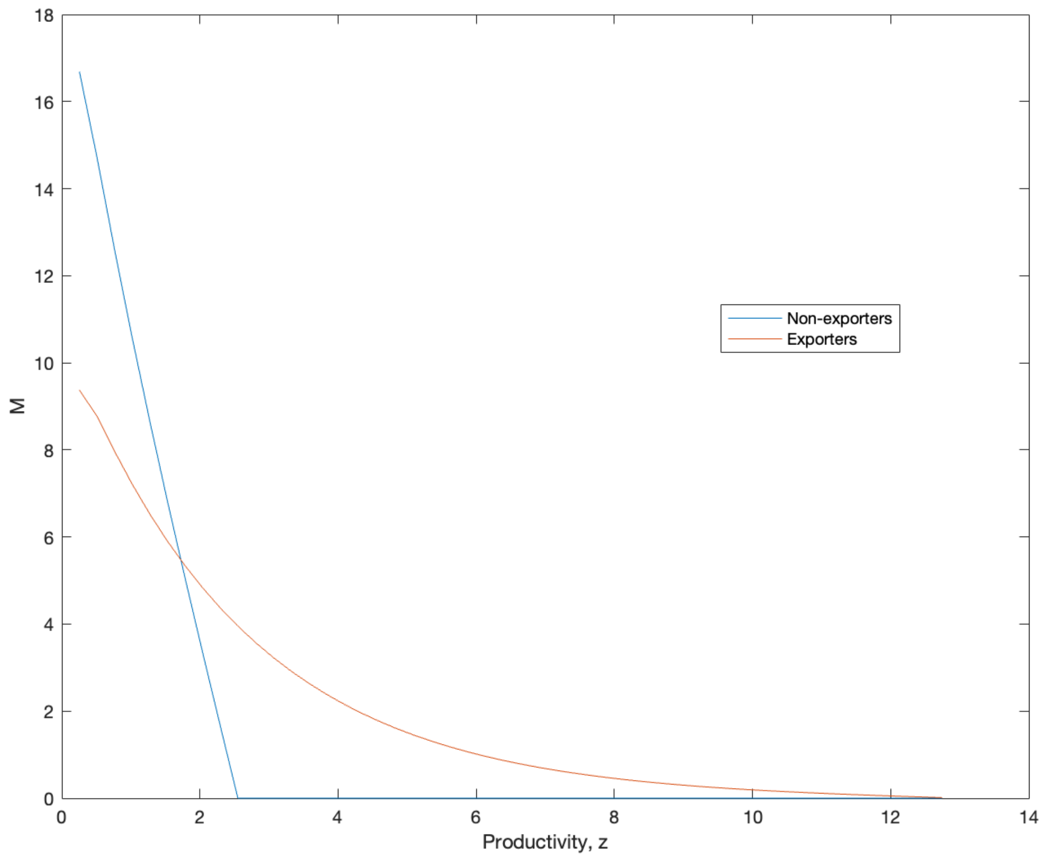

Figure 2: Measure of exporters/non-exporters

Average size of non-exporters that become exporters relative to the average size of exporters that become exporters is given by:

$$\frac{\bar{q}M_n(z_x - \Delta z)\exp(z_x - \Delta z)}{(1-\bar{q})M_x(z_n + \Delta z)\exp(z_n + \Delta z)(1 + \tau^{(1-\rho)})}$$

Question 7

Facts 1-4 in Alessandria-Arkolakis-Ruhl (2020)'s Annual Review paper:

1. **Past export participation is the main predictor of current export participation.**

   The productivity cutoff z_x implies that exporters have higher productivity than non-exporters. Since the decision to export is monotone in productivity, and the exogenous productivity shock only affects z by a small amount $\triangle z$, the model would predict that the probability of continuing as an exporter/non-exporter would be larger than otherwise.

2. **Exporter exit rates fall with past export intensity and time in the export market.**

   The model does not have export intensity, it only features the decision to export (i.e. extensive margin of export). Hence it cannot explain Fact #2.

3. **The exporter entry rate is low but is increasing in size and past export activity.**

   The model would explain the first part of Fact #3 (entry rate is increasing in size), as firm employment (and size) are increasing in productivity and the export decision is monotone in productivity. However, in this model, past export activity will not affect the exporter entry rate. For entrants, there is no past export activity. An exporter that stops exporting must incur these sunk costs again to restart exporting, so the exporter entry rate depends only on the cutoff z_x.

4. **Export intensity rises with time in the export market.**

   The model does not explain Fact #4, for the same reason as explained in Fact #2.

Matlab Code

```
% Econ 282B Homework 1
% Ekaterina Gurkova, Christopher Saw
% Winter 2022

clear;
%% Parameters
global R beta delta rho tau q dz fe f fx fxs N theta z_min z_max z_n z_x
R = 1.05;
beta = 1/R;
delta = 0.005;
rho = 5;
tau = 1.231;
q = 0.5;
dz = 0.25;
fe = 1;
f = 0.1;
fx = 1.4;
fxs = 10;

% Distribution of productivity
N = 50;
z_min = 0.25;
z_max = z_min+dz*N;
theta = 5;


% Grid for productivity
z = linspace(z_min-dz, z_max+dz, N+2);

% Initial guess for market demand index
%Pi_d_min = fx/(exp(z_min)*tau^(1-rho));
%Pi_d_max = (fx+fxs)/(exp(z_max)*tau^(1-rho));
Pi_d_min = 0;
Pi_d_max = 5;
Pi_d = (Pi_d_min+Pi_d_max)/2;


%% Iterating over Pi_d
max_iter = 100000;
iter = 1;
tol = 0.0001;
err = 1;
abs1 = 1;
abs2 = 1;

```

```
for i=1:N+2
    G(i)=(theta*z_min)^theta / z(i)^(theta+1);
end

Vn = linspace(0,0,N);
Vx = linspace(0,0,N);
Vn_new = linspace(0,0,N);
Vx_new = linspace(0,0,N);
while (abs(err) >= tol) && (iter <= max_iter)
    disp(iter);
    % Value function iteration
    while ((abs1 >= tol) || (abs2 >= tol)) && (iter <= max_iter)
        % Solving for productivity cutoffs for exporting/non-exporting
        z_n = log(fx) - log(Pi_d) - (1-rho)*log(tau);
        z_x = log(fx+fxs) - log(Pi_d) - (1-rho)*log(tau);

        % Distribution of productivity
        %pn = (dz/(z_max-z_min)) / (1-(z_n-z_min)/(z_max-z_min));
        %px = (dz/(z_max-z_min)) / ((z_x-z_min)/(z_max-z_min));
        pn = (-(z_min/(z_n+dz))^theta+((z_min/(z_n))^theta))...
            /((z_min/z_n)^theta);
        px = (-(z_min/(z_x))^theta+((z_min/(z_x - dz))^theta))...
            /(1-(z_min/z_x)^theta);


        for i = 2:N-1
            Vx_new(i) = (1+tau^(1-rho))*Pi_d*exp(z(i)) - f - fx  -...
                fxs + beta*(1-delta)*(q*(Vx(i+1) + fxs) + ...
                (1-q)*((1-pn)*(Vx(i-1)+fxs)+pn*Vn(i-1)));
            Vn_new(i) = Pi_d*exp(z(i)) - f + beta*(1-delta)*...
                (q*(px*Vx(i+1)+(1-px)*Vn(i+1))+(1-q)*Vn(i-1));
        end

        Vx_new(1) = (1+tau^(1-rho))*Pi_d*exp(z(1)) - f - fx - fxs +...
            beta*(1-delta)*(q*(Vx(2) + fxs) + (1-q)*0);
        Vn_new(1) = Pi_d*exp(z(1)) - f + beta*(1-delta)*...
                (q*(px*Vx(2)+(1-px)*Vn(2))+(1-q)*0);

        Vx_new(N) = (1+tau^(1-rho))*Pi_d*exp(z(N)) - f - fx - fxs +...
            beta*(1-delta)*(q*(Vx(N) + fxs) + (1-q)*((1-pn)*(Vx(N-1)
               +...
            fxs)+pn*Vn(N-1)));
        Vn_new(N) = Pi_d*exp(z(N)) - f + beta*(1-delta)*...
                (q*(px*Vx(N)+(1-px)*Vn(N))+(1-q)*Vn(N-1));

        Vn_new(Vn_new < 0) = 0;
        Vx_new(Vx_new < 0) = 0;
```

```
        abs1 = max(abs(Vn_new - Vn));
        abs2 = max(abs(Vx_new - Vx));
        Vn = Vn_new;
        Vx = Vx_new;
    end


    for i=1:N
        if z(i)< z_x
            EVN(i) = Vn(i)*G(i+1);
            EVX(i) = 0;
        else
            EVN(i) = 0;
            EVX(i) = Vx(i)*G(i+1);
        end
    end
    EVX_sum = sum(EVX);
    EVN_sum = sum(EVN);


    if beta*(EVN_sum + EVX_sum) - fe > 0
        Pi_d_max = Pi_d;
    else
        Pi_d_min = Pi_d;
    end

    Pi_d = (Pi_d_max + Pi_d_min)/2;
    err = Pi_d_max - Pi_d_min;
    iter = iter+1;

end

%% Mass of firms
exit = zeros(N,1);
exit(Vn>0) = 1;
z_exit = find(exit,1);


err1 = 1;
iter1 = 1;
Mn = linspace(0,0,N);
Mx = linspace(0,0,N);
Mn_new = linspace(0,0,N);
Mx_new = linspace(0,0,N);
while (err1 >= tol) && (iter1 <= max_iter)

    for i=1:N
```

```
            ln(i) = (rho-1)*Pi_d*exp(z(i+1))*Mn(i);
            lx(i) = (rho-1)*Pi_d*(1+tau^(1-rho))*exp(z(i+1))*Mx(i);
        end
        Lp_Me = sum(ln) + sum(lx);
        M_e = 1/(Lp_Me + fe +f*sum(Mn) + (f+fx+fxs)*sum(Mx));

        for i=2:N-1
            if (z(i+1) > z_n) && (z(i+1) ~= z_x)
                Mx_new(i) = M_e*G(i+1) + (1-delta)*(q*Mx(i-1)+(1-q)*Mx(i+1));
            elseif z(i+1) == z_x
                Mx_new(i) = M_e*G(i+1) + (1-delta)*(q*(Mx(i-1)+Mn(i-1)) ...
                    +(1-q)*Mx(i+1));
            else
                Mx_new(i) = 0;
            end

            if (z(i+1) > z(z_exit)) && (z(i+1) < z_x) && (z(i+1) ~= z_n)
                Mn_new(i) = M_e*G(i+1) + (1-delta)*(q*Mn(i-1)+(1-q)*Mn(i+1));
            elseif z(i+1) == z_n
                Mn_new(i) = M_e*G(i+1) + (1-delta)*(q*Mn(i-1)+(1-q) ...
                    *(Mx(i+1)+Mn(i+1)));
            else
                Mn_new(i) = 0;
            end
        end

        if (z(2) > z_n)
                Mx_new(1) = M_e*G(2) + (1-delta)*(1-q)*Mx(i+1);
        else
                Mx_new(1) = 0;
        end

        if (z(2) > z(z_exit)) && (z(2) < z_x) && (z(2) ~= z_n)
                Mn_new(1) = M_e*G(2) + (1-delta)*(1-q)*Mn(2);
        elseif z(2) == z_n
                Mn_new(1) = M_e*G(2) + (1-delta)*(1-q)*(Mx(2)+Mn(2));
        else
                Mn_new(1) = 0;
        end


        if (z(N+1) > z_n) && (z(N+1) ~= z_x)
            Mx_new(N) = M_e*G(N+1) + (1-delta)*q*Mx(N-1);
        elseif z(N+1) == z_x
            Mx_new(N) = M_e*G(N+1) + (1-delta)*q*(Mx(N-1)+Mn(N-1));
        else
```

```
            Mx_new(N) = 0;
        end

        if (z(N+1) > z(z_exit)) && (z(N+1) < z_x)
            Mn_new(N) = M_e*G(N+1) + (1-delta)*q*Mn(N-1);
        else
            Mn_new(N) = 0;
        end




        err1 = max(max(abs(Mn_new-Mn)), max(abs(Mx_new-Mx)));
        iter1 = iter1+1;
        Mn = Mn_new;
        Mx = Mx_new;
end


plot(z(2:N+1),log(ln));
hold on;

plot(z(2:N+1),log(lx));
hold off;
```