A Simulation of Economic Equilibrium with Consideration of the Capital Good

Name:
Joy Jiang
Yifei Jiang

Professor:
Charles S. Peskin

Joy Jiang: zj780@nyu.edu
Yifei Jiang: yj2176@nyu.edu

Modeling and Simulation in Science, Engineering, and Economics

December 16, 2022

Contents

1 Introduction

Economic equilibrium is a state of the balance of economic forces, so the dynamics of economic growth is a question that deserves study. To study the dynamics of economic growth, we follow the evolution of the model over time, including the states before and after the balance is achieved [2].

Under the economic growth model proposed by von Neumann [1], we should find that over time selection, the unproductive processes will be eliminated in favor of the more productive ones and some goods will be eliminated by becoming the free goods [2]. By adding the capital good to modify von Neumann's model, we use the continuous time instead of the discrete time.

Since many parameters will affect the final equilibrium state, we perform the parameter analysis by changing the variance of the need for capital equipment of each process to see what will happen to the economic growth. We make the plot of the economic growth rate under different variances. The detailed description and results are shown in Section 5.

2 Equations

The model in this report is mainly based on Xu and Peskin's paper [3]. Since the capital good is considered in the model, the model is set up based on continuous time, which is modeled by the differential equations.

We consider the economy with n processes indexed by $i = 1, ..., n$, and $m + 1$ goods indexed by $j = 0, 1, ..., m$. Here the 0^{th} good is the capital equipment.

Let π be an $n \times (m + 1)$ matrix which looks like

$$\begin{bmatrix} \pi_{10} & \pi_{11} & \cdots & \pi_{1m} \\ \pi_{20} & \pi_{21} & \cdots & \pi_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \pi_{n0} & \pi_{n1} & \cdots & \pi_{nm} \end{bmatrix}$$

and c be an $1 \times n$ matrix which looks like

$$\begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}$$

Here π_{ij} and c_i are given constants characterizing the economy. We assume the elements in π follow the standard normal distribution and the elements in c follow the gamma distribution.

Let $r_i(t)$ denote the rate at which process i is running at time t, which is also named the intensity. Let $p_j(t)$ denote the price of the good j at time t.

Assume that every process maintains a balanced budget at every time t, which means

$$c_i p_0(t) \frac{dr_i}{dt}(t) = r_i(t) \sum_{j=0}^{m} \pi_{ij} p_j(t). \qquad i = 1, ..., n \tag{1}$$

Since only the relative prices matter, we set $p_0(t) = 1$ and then we can write Eq. 1 in this way:

$$c_i \frac{dr_i}{dt}(t) = r_i(t)(\pi_{i0} + \sum_{j=1}^{m} \pi_{ij} p_j(t)). \qquad i = 1, ..., n \tag{2}$$

We define the excess demand for good j at time t, denoted $e_j(t)$, be the rate at which good j is being consumed minus the rate at which good j is being produced. After performing some simplification, we have

$$e_0(t) = \sum_{i=1}^{n} r_i(t) \sum_{k=1}^{m} \pi_{ik} p_k(t) \tag{3}$$

$$e_j(t) = -\sum_{i=1}^{n} r_i(t) \pi_{ij}. \qquad j = 1, ..., m \tag{4}$$

Then we are interested in two kinds of price equilibrium denoted as $PE1$ and $PE2$. The constraints are stated as followed:

- PE1

$$p_j(t) \geq 0. \qquad j = 1, ..., m. \qquad t \geq 0$$
$$e_j(t) \leq 0. \qquad j = 0, ..., m. \qquad t \geq 0$$

  By applying the Walras' Law, we should find

$$e_0(t) = 0. \qquad t \geq 0$$

$$e_j(t) p_j(t) = 0. \qquad j = 1, ..., m. \qquad t \geq 0$$

- PE2

$$e_j(t) = 0. \qquad j = 1, ..., m. \qquad t \geq 0$$

  By applying the Walras' Law, we should find

$$e_0(t) = 0. \qquad t \geq 0$$

Let $C(t)$ denote the total amount of capital in the economy at time t, which is:

$$C(t) = \sum_{i=1}^{n} c_i r_i(t) \tag{5}$$

Let $\alpha_i(t)$ denote the fraction of capital held by process i at time t, which is:

$$\alpha_i(t) = \frac{c_i r_i(t)}{C(t)} \tag{6}$$

Let $S(t)$ denote the entropy of the economy, which is:

$$S(t) = -\sum_{i=1}^{n} \alpha_i(t) \log \alpha_i(t) \tag{7}$$

So we can calculated S_{max} in this way:

$$S_{max} = -\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n$$

Let $D(t)$ denote the diversity index of the economy, which is:

$$D(t) = \frac{S(t)}{S_{max}} \tag{8}$$

Let $G(t)$ denote the relative growth rate of the economy, which is:

$$G(t) = \frac{1}{C(t)} \frac{dC}{dt}(t) \tag{9}$$

3 Numerical Method

By applying Euler's method to Eq. 2, we have

$$c_i \frac{r_i(t+\Delta t) - r_i(t)}{\Delta t} = r_i(t)(\pi_{i0} + \sum_{j=1}^{m} \pi_{ij} p_j(t)) \tag{10}$$

which is

$$r_i(t+\Delta t) = r_i(t)(1 + \frac{\Delta t}{c_i}(\pi_{i0} + \sum_{j=1}^{m} \pi_{ij} p_j(t))) \tag{11}$$

Similarly, we have

$$e_j(t+\Delta t) = e_j(t) - \Delta t(b_j(t) + \sum_{k=1}^{m} A_{jk}(t) p_k(t)) \tag{12}$$

for $j = 1, ..., m$ where

$$A_{jk}(t) = \sum_{i=1}^{n} \frac{r_i(t)}{c_i} \pi_{ij} \pi_{ik}. \qquad j, k = 1, ..., m \tag{13}$$

$$b_j(t) = \sum_{i=1}^{n} \frac{r_i(t)}{c_i} \pi_{i0} \pi_{ij}. \qquad j = 1, ..., m \tag{14}$$

We define

$$\phi(p, t) = \frac{1}{2} p^T A(t) p + p^T (b(t) - \frac{1}{\Delta t} e(t)) \tag{15}$$

So we can have

$$e_j(t + \Delta t) = -\Delta t (\nabla \phi)(p(t), t) \tag{16}$$

Therefore, to find the price under PE1, we need to minimize $\phi(p, t)$ with constraint $p(t) \geq 0$. Also, to find the price under PE2, we need to minimize $\phi(p, t)$ with no constraints.

4 Validation

4.1 Check the Time Step

As shown in Fig. 1, by fixing the total simulation time, we change the time step dt from 0.0001 to 0.0002 and we find that the results are almost the same, which shows that for dt small enough, it will not affect the result.

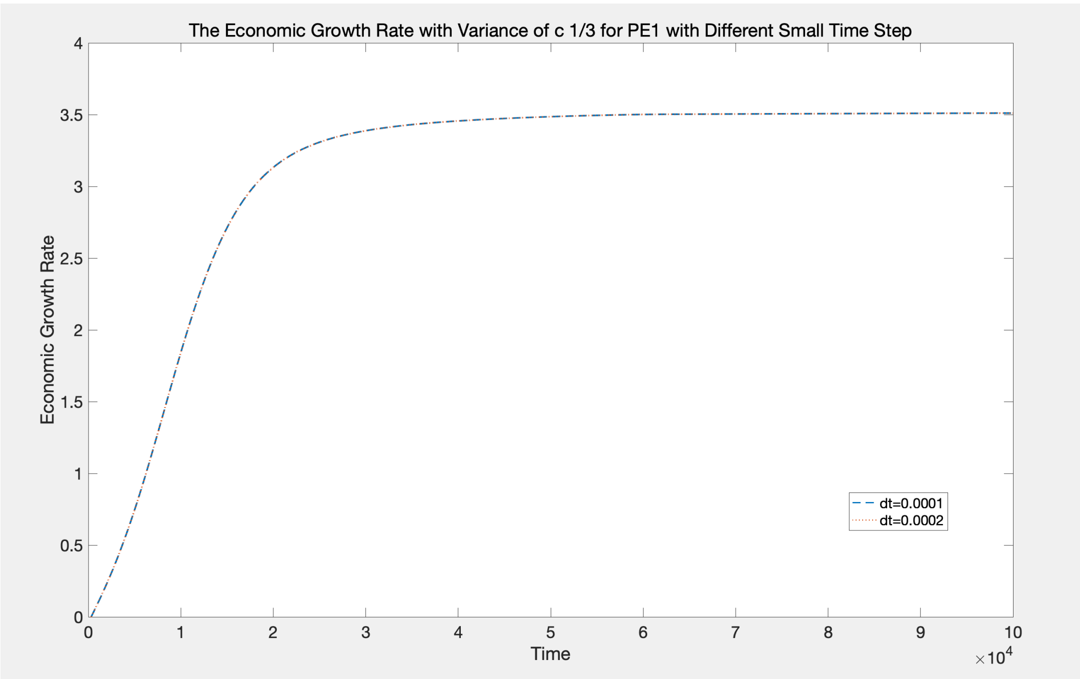

Figure 1: The Economic Growth Rate under Different dt

Since the code for simulating PE1 and PE2 are quite similar, we only show the validation result for PE1, but it is sure that we will get the same result if we do the same for PE2.

4.2 Check the Excess Demand

We plot the value of e_j ($j = 0$ for the capital good and $j \geq 1$ for non-capital goods) which is shown in Fig. 2 and Fig. 3.

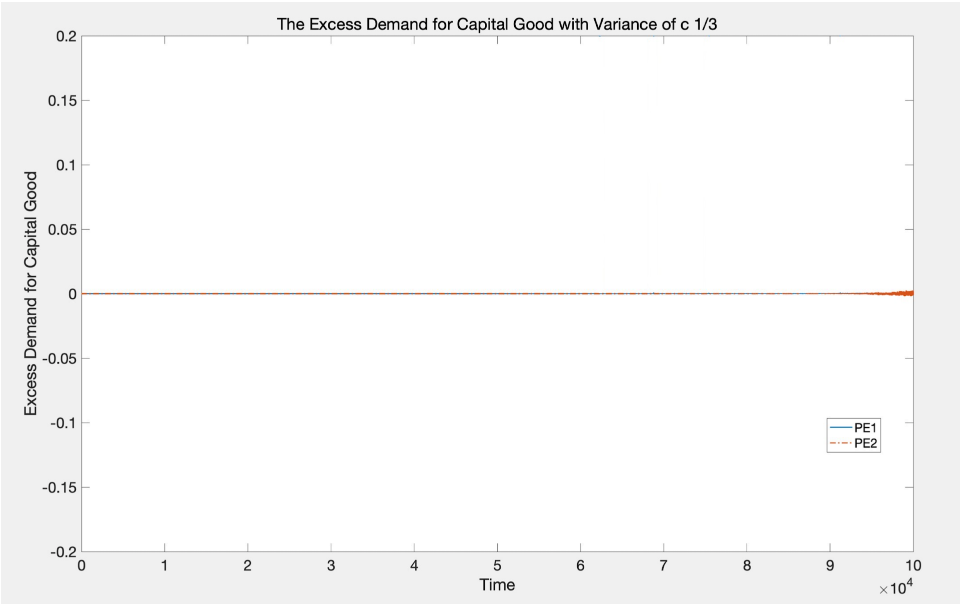

Figure 2: The Excess Demand for Capital Good

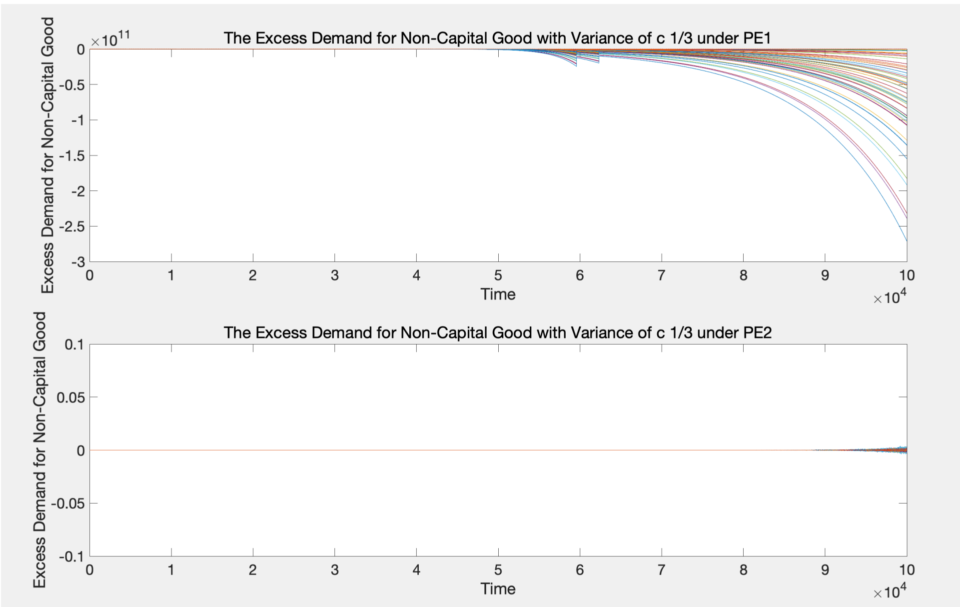

Figure 3: The Excess Demand for Non-Capital Good

We see that for both PE1 and PE2, $e_0(t)$ is very close to 0, which is consistent with our condition. Also, for $e_j(t)$ with $j = 1, ..., m$, it is non-positive under PE1 and almost equal to 0 under PE2, which is also consistent with our condition.

5 Results and Discussion

During the simulation of the economic problem with the capital good with MATLAB, we set $n = 1000$ and $m = 100$. The detailed code is shown in the Appendix.

By performing the simulation, we are able to track the trends of prices $p_j(t)$, intensities $r_i(t)$, the diversity index $D(t)$ of the economy, and the relative growth rate $G(t)$ under PE1 and PE2 settings respectively.

Recall that we can vary the differences of how much capital is needed among the processes by changing the variance of the c which follows the gamma distribution. In order to find out if the "capital needs distribution across processes" has a correlated effect on "economic growth", we conducted 4 pairs of experiments with different variance initialization on c, and fixing all other parameters and variables the same. Specifically, we let the mean of c be 1, and the variance of c be $\frac{1}{9}$, $\frac{1}{5}$, $\frac{1}{3}$, $\frac{1}{2}$ in each case, and then compare the results as follows.

5.1 Price

We calculate the price of m goods over time under PE1 and PE2 scenarios by minimizing $\phi(p, t)$ shown in Eq. 15. The results are shown in Fig. 4 and Fig. 5.

Under PE1, the price is only allowed to be non-negative, so that when the price gets to zero, it would then either jump up to the positive or become the free good forever; but in PE2, we do see the prices go to negative under the market clearing setting. When we compare the plots across our 4 pairs of experiments, we are not able to see a direct correlation between the variance of c and the price changes. So we conclude that the variance of c will not affect the final price of each non-capital good.

5.2 Intensity

We update the intensity of n processes over time under PE1 and PE2 scenarios with Eq. 11. The results of log scale plot are shown in Fig. 6 and Fig. 7.

The slope of log(r) in terms of t presents the growth rate. From both of the graphs in PE1 and PE2, we can see that the growth rates are smoothly distributed. If we compare the distributions between our 4 pairs of experiments, we find that the lines tend to scatter across a greater range over time when the variance of c is greater.

As shown in the following Fig. 8 and Fig. 9, we plot the distribution of intensities of n processes at the last time step. Under both PE1 and PE2 scenarios, the distributions of r seem to become more scattered as the variance of c gets larger. In other words, in most cases, when the difference of capital needs across processes is larger, the difference of intensities across processes is also larger.

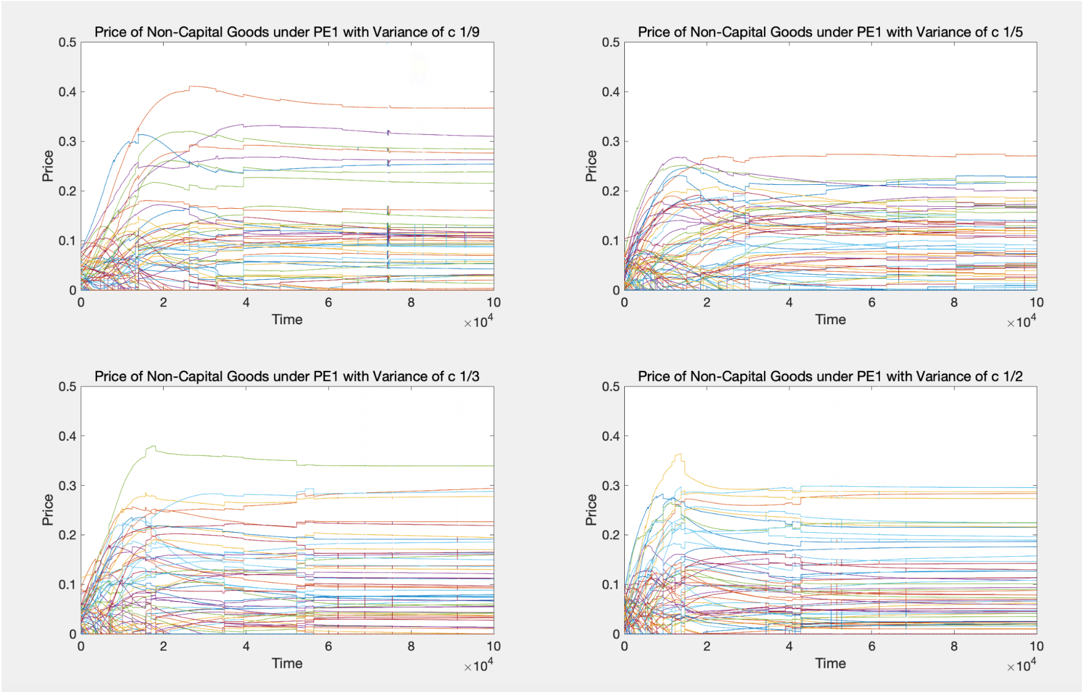

Figure 4: Prices under PE1 Scenario

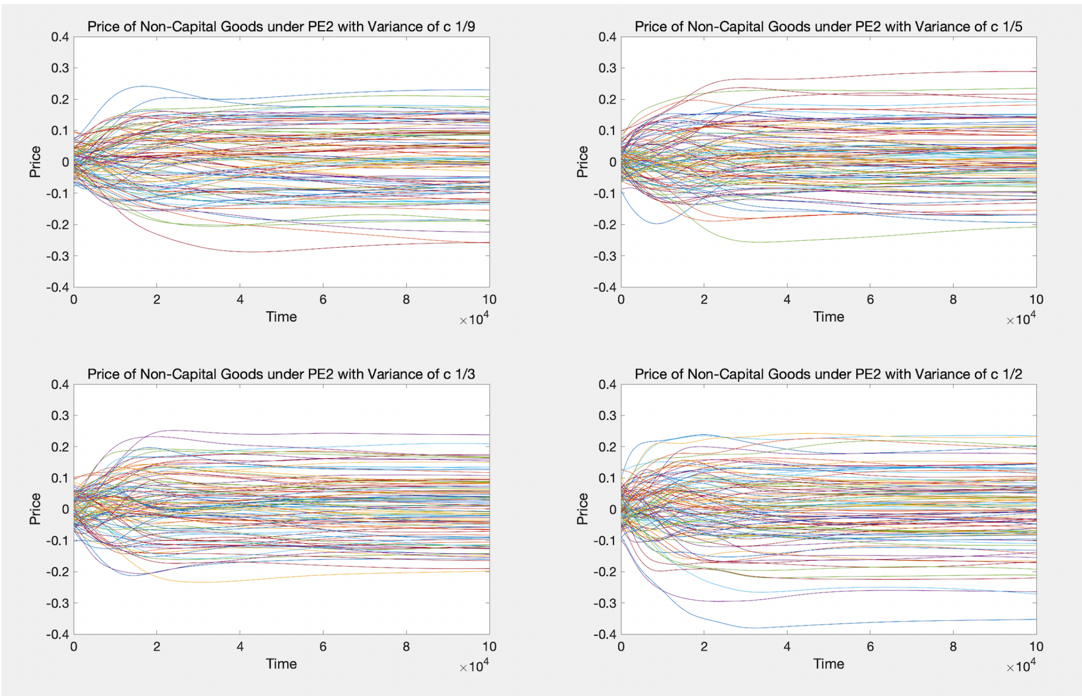

Figure 5: Prices under PE2 Scenario

5.3 Diversity

We define the diversity index with entropy, which describes the amount of capital that each process is holding. We calculate the diversity with Eq. 8 and the results are shown in Fig. 10 and Fig. 11.

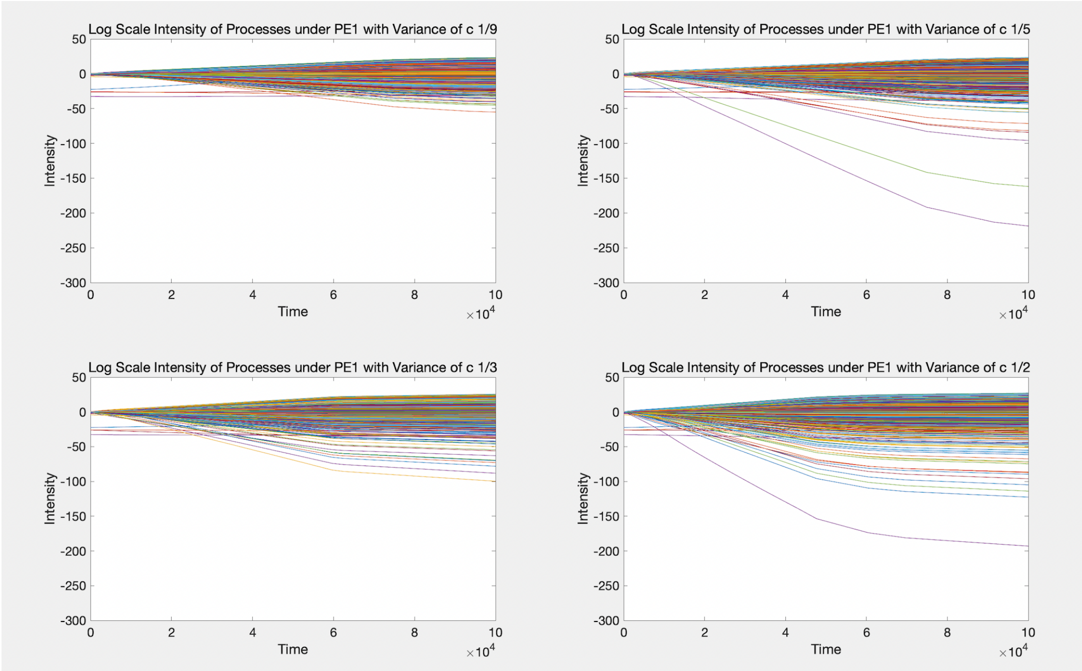

Figure 6: Log scale intensities under PE1 Scenario

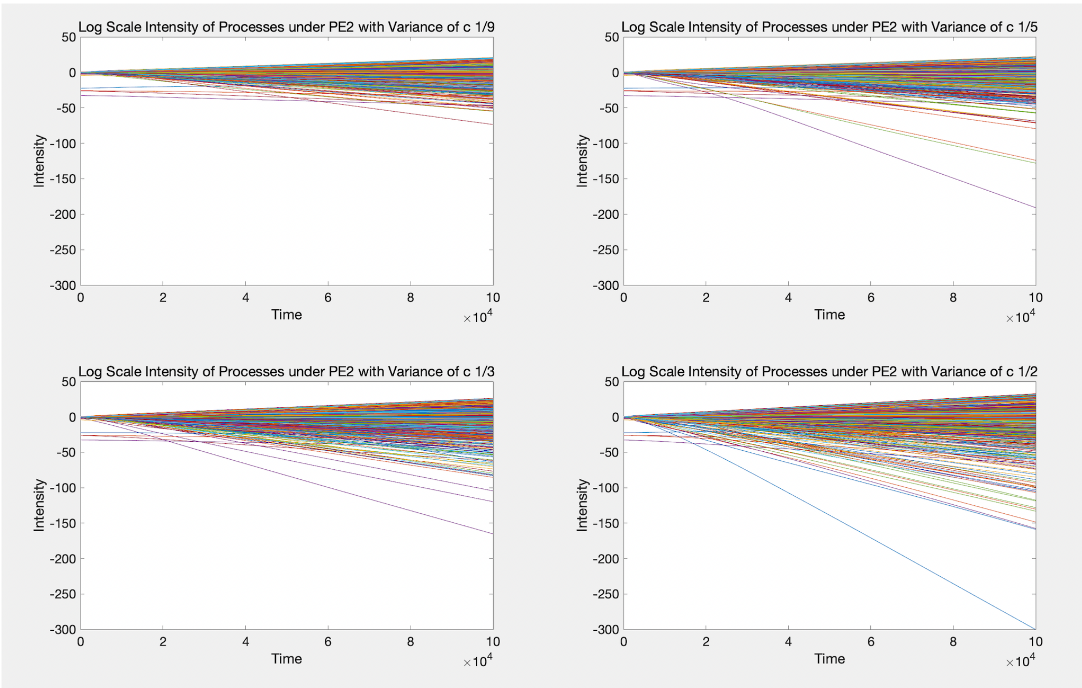

Figure 7: Log scale intensities under PE2 Scenario

In Fig. 10 and Fig. 11, we can see that, under PE2, as the economy approaches Balanced-Growth, the diversity of the economy is less when the variance of c is greater, meaning that the capital is concentrated among specific processes. Under PE1, while the diversity trend of the economy is similar when approaching Balanced-Growth, the diversity value is not always negatively correlated with c's variance as we expect. We see the diversity curve crosses over as the time gets very large. More attention and analysis are needed when this particular case raises.

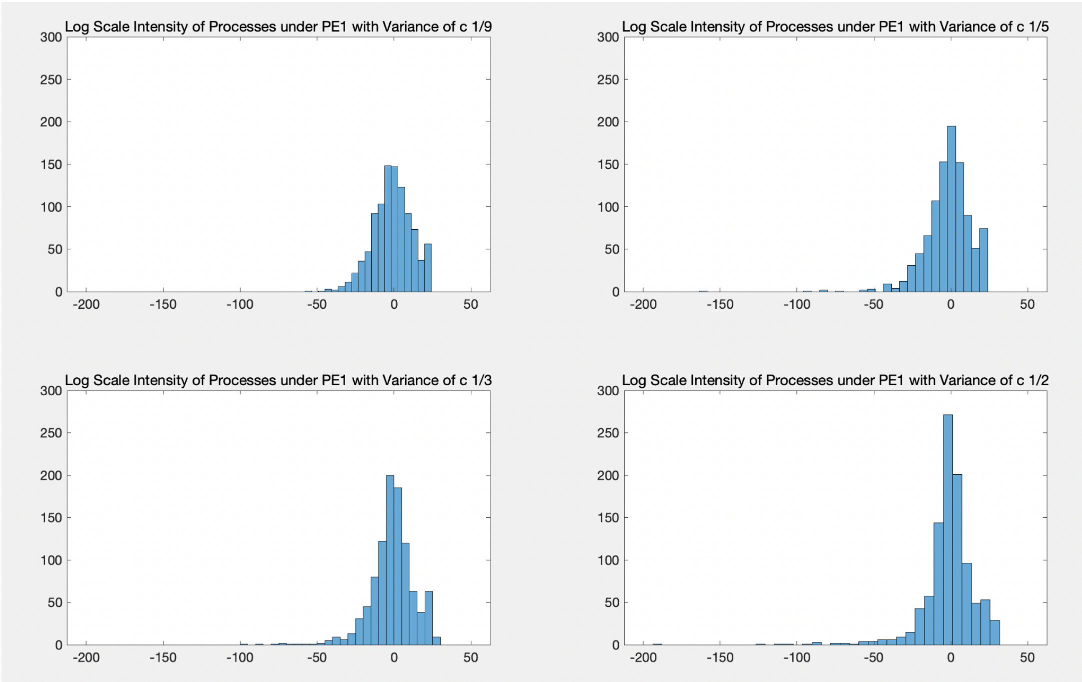

Figure 8: Intensity distribution at the last t under PE1 Scenario

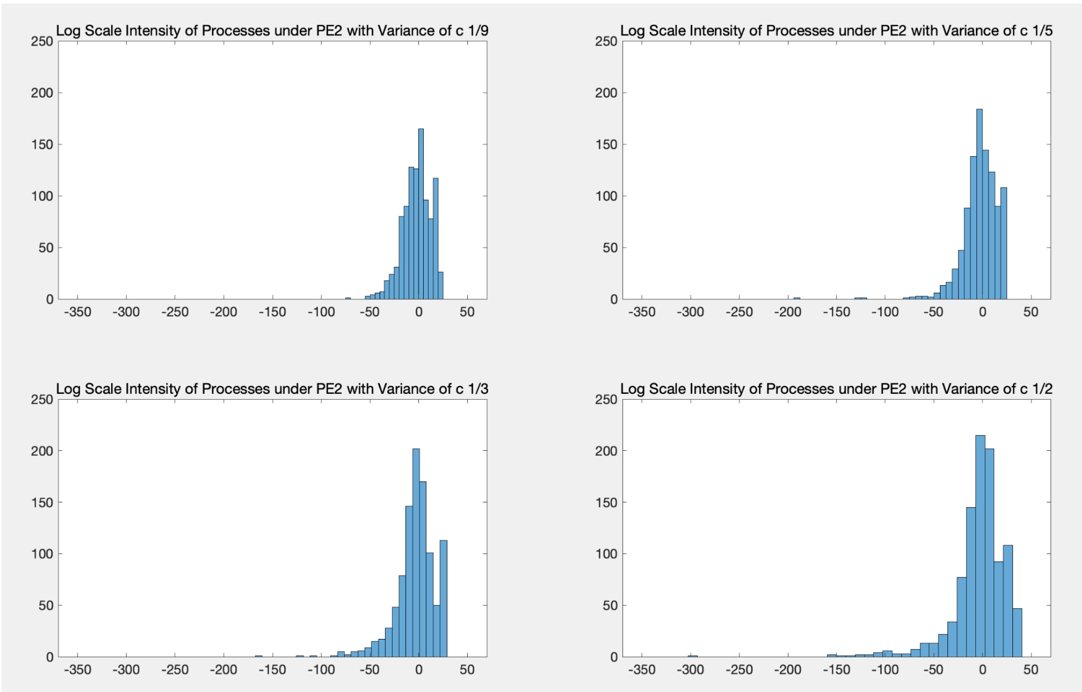

Figure 9: Intensity distribution at the last t under PE2 Scenario

5.4 Economic Growth

By calculating the relative growth rate G in terms of t with Eq. 9, we are able to simulate the economic growth performance here.

In Fig. 12 and Fig. 13, we see that both the models of PE1 and PE2 reach the

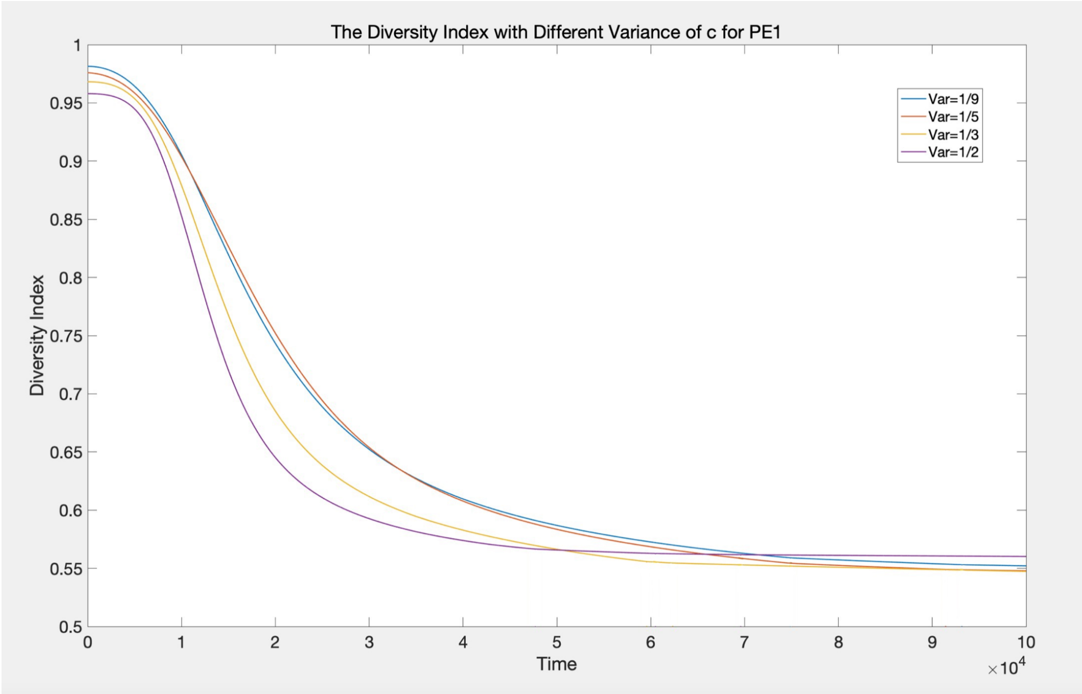

Figure 10: Diversity index under PE1 Scenario

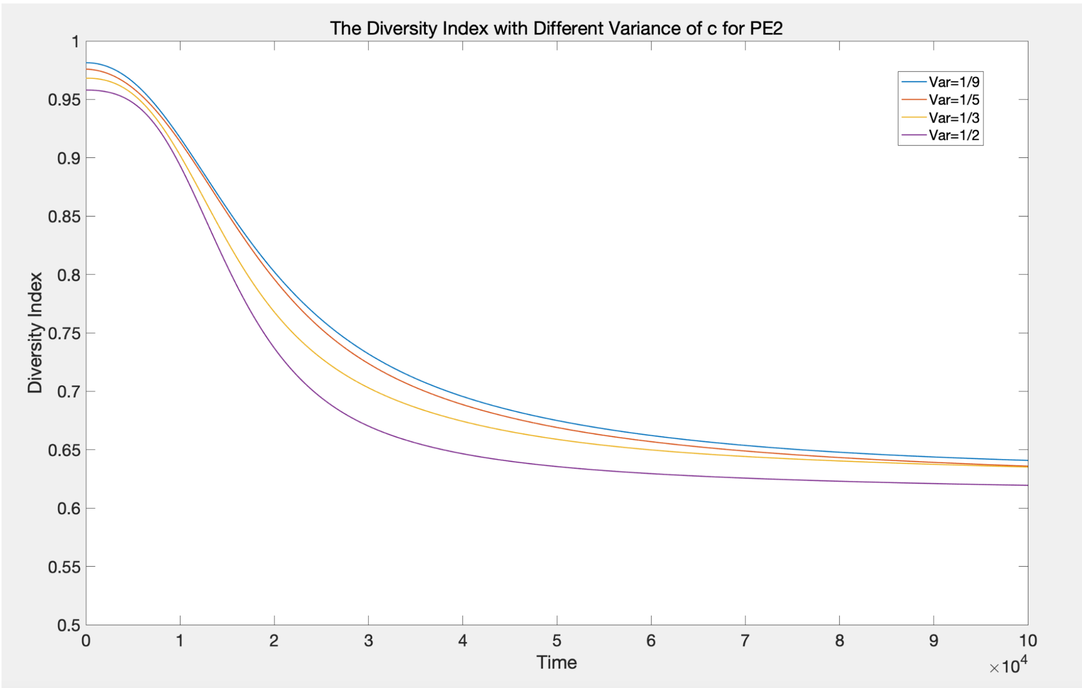

Figure 11: Diversity index under PE2 Scenario

Balanced-Growth over time. By comparing the growth curves in each experiment case, we discover that the economy attains a higher relative growth rate with a greater variance of c. When the capital needs of each process are less evenly distributed, the economy is more likely to have a better performance in terms of growth. While it is not confirmed that the variance in capital needs would cause a higher economic growth rate, we say that it values to consider this factor as a relevant indicator when

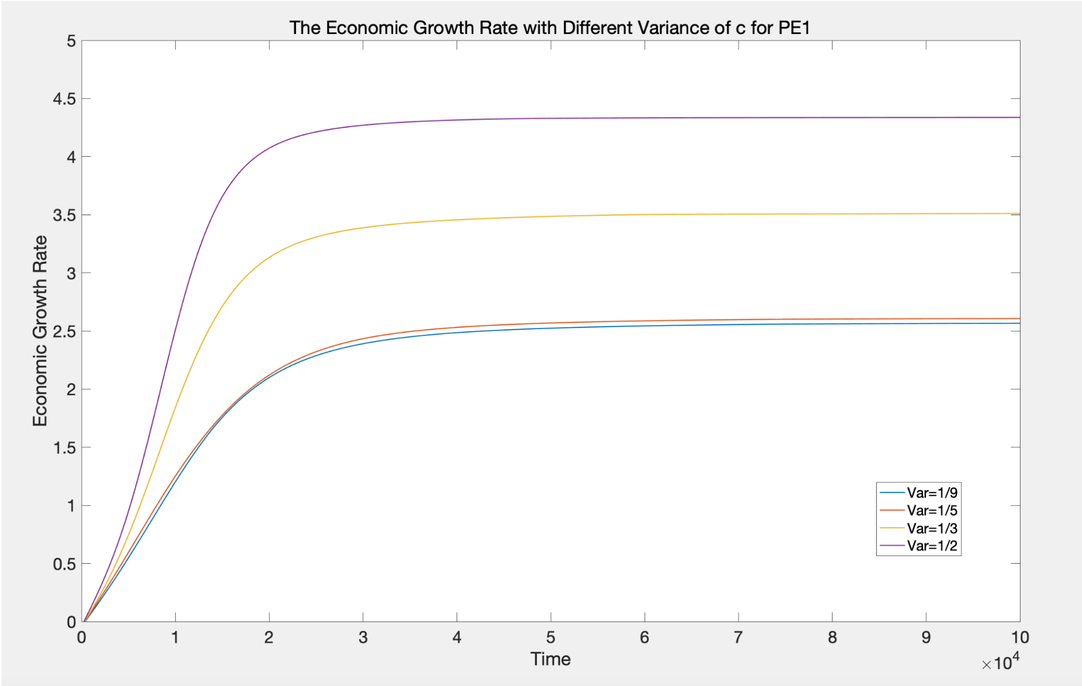

Figure 12: Economic Growth Rate under PE1 Scenario

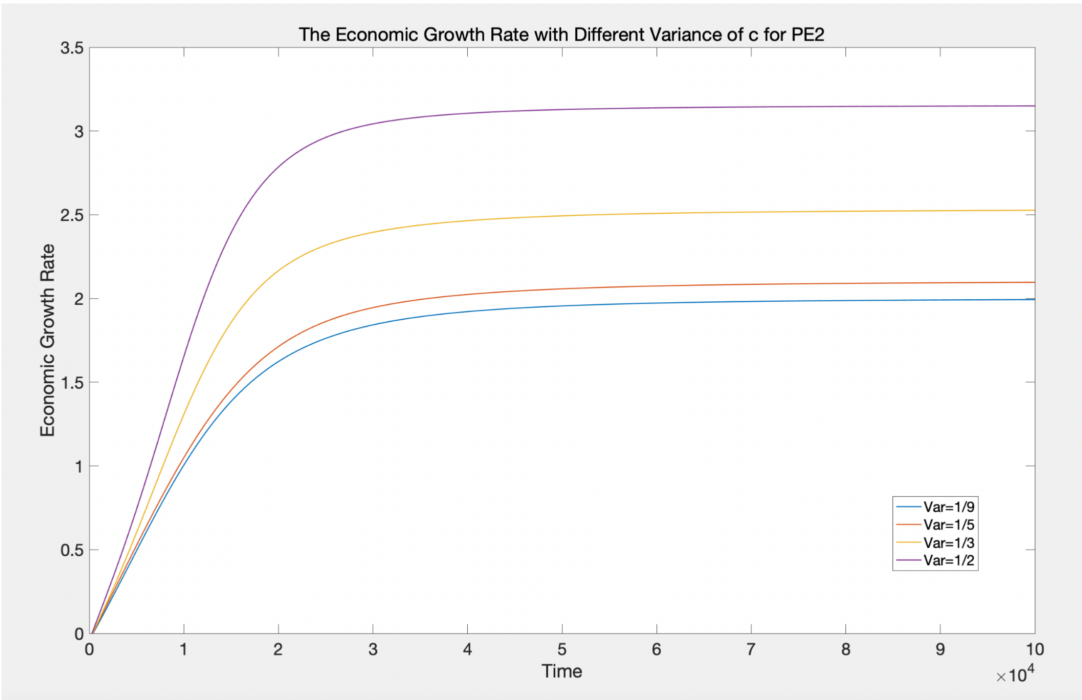

Figure 13: Economic Growth Rate under PE2 Scenario

evaluating or analyzing an economy.

6 Summary and Conclusions

In this project, we have constructed and simulated the growth of an economy based on von Neumann's model with the inclusion of capital. From the results we derived, the economy is observed to reach Balanced-Growth as von Neumann's original model proposed.

Based on our analysis of capital variance variations, we conclude that the extent to which how processes differ in capital needs has a positive relationship with economic growth rate. Although we are not able to propose a causal effect, this correlation may be included as an indicating factor in economic analysis.

References

[1] J v Neumann. A model of general economic equilibrium. *The Review of Economic Studies*, 13(1):1–9, 1945.

[2] Charles S. Peskin. Notes on economic growth and price equilibrium. 2019.

[3] Shubo Xu and Charles S Peskin. The impact of universal recycling on the evolution of economic diversity. *PloS one*, 17(1):e0262184, 2022.

Computer code as an Appendix

A generate.m

```
% generate pi and c

clc, clear all, close all

m = 100;
n = 1000;

% the members in pi follows the standard normal distribution
pi = randn([n,m+1]);

% the members in c follows the gamma distribution
% change the variance of c
c_1over2 = gamrnd(2,(1/2),[1,n]);  % mean=1, var=1/2
c_1over3 = gamrnd(3,(1/3),[1,n]);  % mean=1, var=1/3
c_1over5 = gamrnd(5,(1/5),[1,n]);  % mean=1, var=1/5
c_1over9 = gamrnd(9,(1/9),[1,n]);  % mean=1, var=1/9

% save the data
save("generator.mat",'pi','c_1over2','c_1over3','c_1over5','c_1over9');
```

B Econ_PE1.m

```
% Econ_PE1

clc, clear all, close all

%%
% m = the number of non-capital goods
% n = the number of processes
% T = the number of simulation times
% dt = the time step
% q = the variable to reduce the step size when there is an overflow

% pi = n*(m+1) matrix characterizing the model economy
% c = 1*n matrix characterizing the model economy

% t(index) = the time
% r(index,i) = the intensity of process i at time index
% p(index,j) = the price of good j at time index
% e0(index) = the excess demand of the capital good at time index
% e(index,j) = the excess demand of the non-capital good j at time ...
    index
% C(index) = the total amount of capital in the economy at time index
% G(index) = the relative growth rate of the economy at time index
% D(index) = the diversity index of the economy at time index

% alpha(i) = the fraction of capital held by process i
% S = the entropy of the economy
%%

% Parameters
```

```
m = 100;
n = 1000;
T = 100000;
dt = 0.0001;
q = 1;

% load the fixed pi and c
% we change the varance of c by loading different variables in ...
    generator.mat
pi = cell2mat(struct2cell(load("generator.mat","pi")));
c = cell2mat(struct2cell(load("generator.mat","c_1over3")));

% initialize all the varibles interested
t = zeros(1,1);
r = zeros(1,n);
p = zeros(1,m);
e0 = zeros(1,1);
e = zeros(1,m);
C = zeros(1,1);
G = zeros(1,1);
D = zeros(1,1);

% get the intensity at time 0
pi_new = pi(:,2:m+1);  % drop the column of capital good
u = ones(1,n);
H = eye(n);
f = zeros(1,n);
Aeq = pi_new';
Beq = -pi_new'*u';
lb = (zeros(1,n)-u)';
w = quadprog(H,f,[],[],Aeq,Beq,lb,[]);
r_t = w'+u;

% perform the simulation
for index = 1:T
    t(index) = dt*index;

    % clear the variables of the previous time
    A = zeros(m,m); % A is calculated for minimizing phi
    b = zeros(m,1); % b is calculated to calculate f for ...
        minimizing phi
    e_t = zeros(m,1);
    f = zeros(m,1); % f is calculated for minimizing phi
    alpha = zeros(1,n);

    for j = 1:m
        for k = 1:m
            for i = 1:n
                % calculate A
                A(j,k) = A(j,k) + r_t(i)/c(i)*pi(i,j+1)*pi(i,k+1);
            end
        end

        for i = 1:n
            % calculate b
            b(j)= b(j) + r_t(i)/c(i)*pi(i,1)*pi(i,j+1);
            % calculate e_t
            e_t(j) = e_t(j) - r_t(i)*pi(i,j+1);
```

```
        end
        % calculate f
        f(j) = b(j)-e_t(j)/dt;
    end
    % phi = 1/2 * (p_t.') * A * p_t + (p_t.') * (b - 1/dt * e);
    % we use quadprog as followed to minimize phi (with the ...
       constraint)
    p_t = quadprog(A,f,[],[],[],[],zeros(m,1),[]);

    % calculate e
    e(index,:) = e_t;

    % update p
    p(index,:) = p_t;

    % update r
    for i = 1:n
        r_temp = 0;
        for j = 1:m
            r_temp = r_temp + pi(i,j+1)*p_t(j);
        end
        r_t(i) = r_t(i) * (1 + dt/c(i) * (pi(i,1) + r_temp));
    end
    r(index,:) = r_t;

    % calculate C and G
    dCdt =0;
    C(index) = 0;
    for i = 1:n
        dCdt_temp = 0;
        for j = 1:m
            dCdt_temp = dCdt_temp + pi(i,j+1)*p_t(j);
        end
        dCdt = dCdt + r_t(i)*(dCdt_temp +pi(i,1));
        C(index) = C(index) + r_t(i)*c(i);
    end
    G(index) = dCdt/C(index);
    G(index) = G(index) * q;

    % check and update
    if index > 1
        G_diff = G(index) - G(index-1);
        if G_diff > 1   % the signal for overflow
            pi = pi/2;
            q = q*2;   % update q

            % replace the wrong results of this iteration with the ...
               previous ones
            r_t = r(index-1,:);
            p_t = p(index-1,:);
            r(index,:) = r_t;
            p(index,:) = p_t;
            G(index) = G(index-1);
        end
    end


    % calculate alpha
```

```
        for i = 1:n
            alpha(i) = r_t(i)*c(i)/C(index);
        end

        % calculate S
        S_max =log(n);
        S = 0;
        for i =1:n
            S =  S-alpha(i)*log(alpha(i));
        end

        % calculate D
        D(index) = S/S_max;

        % check e0
        e0(index) = 0;
        for i = 1:n
            e0_temp = 0;
            for k = 1:m
                e0_temp = e0_temp + pi(i,k+1)*p_t(k);
            end
            e0(index) = e0(index) + r_t(i) * e0_temp;
        end

        % check e
        for j=1:m
            e(index,j) = 0;
            for i=1:n
                e(index,j) = e(index,j) - r_t(i)*pi(i,j+1);
            end
        end


    end

    % save the data
    save("PE1_c_var_1over3.mat",'r','c','pi','p','e0','D','G','C','e','t');
```

C Econ_PE2.m

```
% Econ_PE2

clc, clear all, close all

% the explanation of the variables is the same as in Econ_PE1.m

% Parameters
m = 100;
n = 1000;
T = 100000;
dt = 0.0001;
q = 1;

% load the fixed pi and c
% we change the varance of c by loading different variables in ...
    generator.mat
pi = cell2mat(struct2cell(load("generator.mat","pi")));
```

```
c = cell2mat(struct2cell(load("generator.mat","c_1over3")));

% initialize all the varibles interested
t = zeros(1,1);
r = zeros(1,n);
p = zeros(1,m);
e0 = zeros(1,1);
e = zeros(1,m);
C = zeros(1,1);
G = zeros(1,1);
D = zeros(1,1);

% get the intensity at time 0
pi_new = pi(:,2:m+1);  % drop the column of capital good
u = ones(1,n);
H = eye(n);
f = zeros(1,n);
Aeq = pi_new';
Beq = -pi_new'*u';
lb = (zeros(1,n)-u)';
w = quadprog(H,f,[],[],Aeq,Beq,lb,[]);
r_t = w'+u;

% perform the simulation
for index = 1:T
    t(index) = dt*index;

    % clear the variables of the previous time
    A = zeros(m,m); % A is calculated for minimizing phi
    b = zeros(m,1); % b is calculated to calculate f for ...
        minimizing phi
    e_t = zeros(m,1);
    f = zeros(m,1); % f is calculated for minimizing phi
    alpha = zeros(1,n);

    for j = 1:m
        for k = 1:m
            for i = 1:n
                % calculate A
                A(j,k) = A(j,k) + r_t(i)/c(i)*pi(i,j+1)*pi(i,k+1);
            end
        end

        for i = 1:n
            % calculate b
            b(j)= b(j) + r_t(i)/c(i)*pi(i,1)*pi(i,j+1);
            % calculate e_t
            e_t(j) = e_t(j) - r_t(i)*pi(i,j+1);
        end
        % calculate f
        f(j) = b(j)-e_t(j)/dt;
    end
    % phi = 1/2 * (p_t.') * A * p_t + (p_t.') * (b - 1/dt * e);
    % we use quadprog as followed to minimize phi (with no ...
        constraints)
    p_t = quadprog(A,f,[],[],[],[],[],[]);

    % calculate e
```

```
    e(index,:) = e_t;

    % update p
    p(index,:) = p_t;

    % update r
    for i = 1:n
        r_temp = 0;
        for j = 1:m
            r_temp = r_temp + pi(i,j+1)*p_t(j);
        end
        r_t(i) = r_t(i) * (1 + dt/c(i) * (pi(i,1) + r_temp));
    end
    r(index,:) = r_t;

    % calculate C and G
    dCdt =0;
    C(index) = 0;
    for i = 1:n
        dCdt_temp = 0;
        for j = 1:m
            dCdt_temp = dCdt_temp + pi(i,j+1)*p_t(j);
        end
        dCdt = dCdt + r_t(i)*(dCdt_temp +pi(i,1));
        C(index) = C(index) + r_t(i)*c(i);
    end
    G(index) = dCdt/C(index);
    G(index) = G(index) * q;

    % check and update
    if index > 1
        G_diff = G(index) - G(index-1);
    end


    % calculate alpha
    for i = 1:n
        alpha(i) = r_t(i)*c(i)/C(index);
    end

    % calculate S
    S_max =log(n);
    S = 0;
    for i =1:n
        S =  S-alpha(i)*log(alpha(i));
    end

    % calculate D
    D(index) = S/S_max;

    % check e0
    e0(index) = 0;
    for i = 1:n
        e0_temp = 0;
        for k = 1:m
            e0_temp = e0_temp + pi(i,k+1)*p_t(k);
        end
        e0(index) = e0(index) + r_t(i) * e0_temp;
```

```
    end

    % check e
    for j=1:m
        e(index,j) = 0;
        for i=1:n
            e(index,j) = e(index,j) - r_t(i)*pi(i,j+1);
        end
    end


end

% save the data
save("PE2_c_var_1over3.mat",'r','c','pi','p','e0','D','G','C','e','t');
```