Analysis on the Von-Neumann Equilibrium Model

Eddie Park
Edward Fuentes

1 The Von-Neumann Model

The Von-Neumann model of economic growth is based on processes and commodities. There is no differentiation between producers and consumers. This is consistent with the idea that consumers usually also produce their own "commodities" such as labor. There are a few essential components to the model. Let i denote the number of processes and j be the number of commodities(goods) in the economy. Each process is described by two matrices A,B that are independent of time. The A matrix describes how much each process needs to produce each good and the B matrix describes how much each process is asking for each good. In essence, A is the input matrix and B is the output matrix. Unlike the A and B matrices, the price of each good does change with time. The price of good j at time t is denoted by $p_j(t)$. Each process runs at a certain intensity level denoted by R_i where i denotes the process that it belongs to. The intensity rate can be thought of as the rate of production. This rate changes from period to period. Note that unlike the A,B matrices, R_i and P_j do depend on time. We can find the intensity for the next period using:

$$r_i(t+1) = r_i(t)\frac{\sum_{j=1}^{m} B_{ij}p_j(t)}{\sum_{j=1}^{m} A_{ij}p_j(t)}$$

Since it may be the case that more of a certain good is bought than there is available in the market, we must define excess demand. That is given by:

$$e_j(t) = \sum_{i=1}^{n} r_i(t)(\frac{\sum_{k=1}^{m} B_{ik}p_k(t)}{\sum_{k=1}^{m} A_{ik}p_k(t)}A_{ij} - B_{ij})$$

Our focus is on minimizing excess demand using properties of excess demand. First of all, the excess demand function should satisfy Walras' Law which states that the value of excess demand must sum to zero, in our case at any price vector p. It is convenient to show that the excess demand of an idealize economy described by the Von-Neumann model satisfies Walras' Law. To express this, multiply both sides of by p(j), sum over j, and then, on the right-hand side of the equation, bring the sum over j inside the sum over i since the order of multiplication does not matter. In this way, we get:

$$\sum_{j=1}^{m} p_j(t)E_j(t) = \sum_{i=1}^{n} r_i(\frac{\sum_{k=1}^{m} B_{ik}p_k(t)}{\sum_{k=1}^{m} A_{ik}p_k(t)}\sum_{j=1}^{m} A_{ij}p_j(t) - \sum_{j=1}^{m} B_{ij}p_j(t))$$

Since $\sum_{k=1}^{m} A_{ik}p_k(t)$ is essentially $\sum_{j=1}^{m} A_{ij}p_j(t)$, the term inside the parenthesis can be simplify into $\sum_{k=1}^{m} B_{ik}p_k(t) - \sum_{j=1}^{m} B_{ij}p_j(t)$ which is also equal to zero. Therefore, we have proved that our excess demand function satisfies the Walras's Law:

$$\sum_{j=1}^{m} p_j E_j(p) = 0$$

Since the excess demand found satisfies Walras' Law, we can now formulate a function of a given price vector using excess demand such that minimum is 0. We are interested in such a function since its minimum essentially is the price equilibrium vector. To that end, define $\phi(p)$ where $\phi : S \to \mathbb{R}$ as:

$$\phi(p) = \frac{1}{2}\sum_{j:E_j(p)>0} E_j^2(p)$$

We square the excess demand in order to guarantee continuous derivatives which will allows to compute its gradient even at boundary values. The function that we need to minimize over the unit sphere is

$$\frac{\partial\overline{\phi}}{\partial q_k}(q) = \sum_{j:\overline{E}(q)>0} \overline{E_j}(q)\frac{\partial\overline{E_j}}{\partial q_k}(q)$$

where

$$\overline{E}_j(q) = \sum_{i=1}^{n} r_i(\frac{\sum_{k=1}^{m} B_{ik}q_k^2}{\sum_{k=1}^{m} A_{ik}q_k^2}A_{ij} - B_{ij})$$

$\frac{\partial\overline{E_j}}{\partial q_k}(q)$ can be found using Quotient Rule. since r_i,A_{ij} and B_{ij} are considered as constants, we only need to find a derivative of the upper and the lower part of the fraction which are:

$$Derivative\,of\,high = 2B_{ik}q_k$$

$$Derivative\,of\,low = 2A_{ik}q_k$$

and consequently results:

$$\frac{\partial\overline{E}_j}{\partial q_k}(q) = 2q_k\sum_{i=1}^{n} r_i A_{ij}(\frac{B_{ik}\sum_{l=1}^{m} A_{il}q_l^2 - A_{ik}\sum_{l=1}^{m} B_{il}q_l^2}{(\sum_{k=1}^{m} A_{ik}q_k^2)^2})$$

Now that we have all the constituents to solve the problem, we will next consider the gradient descent in terms of an initial value problem to solve the minimization problem. Instead of using an ODE solver in MATLAB, we used Euler's method. In order to ensure our values stay on the unit sphere, we projected the resulted value after each step onto the sphere.

$$\frac{\partial q}{\partial\tau} = -(\nabla\overline{\phi})(q(\tau)),$$

$$q(0) = q^{(0)},$$

$$q(\tau+\triangle\tau) = \frac{q(\tau) - (\triangle\tau)(\nabla\overline{\phi})(q(\tau)))}{\|q(\tau) - (\triangle\tau)(\nabla\overline{\phi})(q(\tau)))\|}$$

2 The Simulation

2.1 General Steps

There are several components to the simulation. The main program consists of two phases: the initial set up of variables and the updating of variables. We took full advantage of the flexibility that comes with using functions in order to update the variables.

During the initial set up we initialize the intensity rate matrix, price matrix and the excess demand matrix. We also randomly assign values between 0 and 1 to the A and B matrices using the rand function and set the rate of intensity for all producers to 1.

The updating phase makes use of several functions. The reason why functions are useful in updating the rate of intensity, price and excess demand is because it allows us to mainly operate in one for loop. Finding the rate of intensity, price and excess demand requires actions such as summing over all producers, summing over all goods, etc. Combined with the fact that we are already in a for loop to control the number of periods the simulation is running for, it adds unnecessary complexity.

First we calculate the intensity rate for the given period using the old periods rate. This done using a function named r̈_updatetḧat takes A,B,Intensity rate matrix, Price matrix and time as inputs. It returns an $ix1$ matrix with the new rate of intensity for each producer. It then adds this column vector to our ixt R matrix. This allows us to index into the R matrix when determining the next rate of intensity or when determining the excess demand.

We then use the intensity rates found in order to calculate the excess demand for each process. This also done using another function named "ex_demand" that takes A,B,Intensity rate matrix, Excess demand matrix and time as inputs. It returns an j x 1 matrix with the excess demand for the given period. We then add this column vector to our excess demand matrix which is j x t. Again, this is useful because we now only have to deal with one matrix which we can index into.

The found excess demand is used to calculate the prices for the next period after minimizing the following function over the unit sphere:

$$\bar{\phi}(q) = \frac{1}{2} \sum_{jE_j(\bar{q})>0} \bar{E_j}^2(q)$$

Once we have the prices, it is crucial that we verify that we have found the true global minimum and not just a local minimum. There are two different methods that we use to verify this. One of them relies on Walras' Law. The excess demand function satisfies Walras' Law:

$$\sum_{j=1}^{m} p_j(t) E_j(p) = 0$$

It is often the case that local minimum also satisfies the Walras' Law. Therefore, we need to verify again by substituting the prices we found into the $\bar{\phi}$ function. Among the minima we found, we select a price vector with a phi value within a certain tolerance of 0.

2.2 The Parameter Optimization

With a high number of time periods, processes, and goods, the speed of nested for loops that our code contains experiences a significant slow down. To avoid using a excessive parameter, such as tau or epsilon in our Euler's method, we implemented parameter tuning. Parameter tuning is often utilized in the field of machine learning.

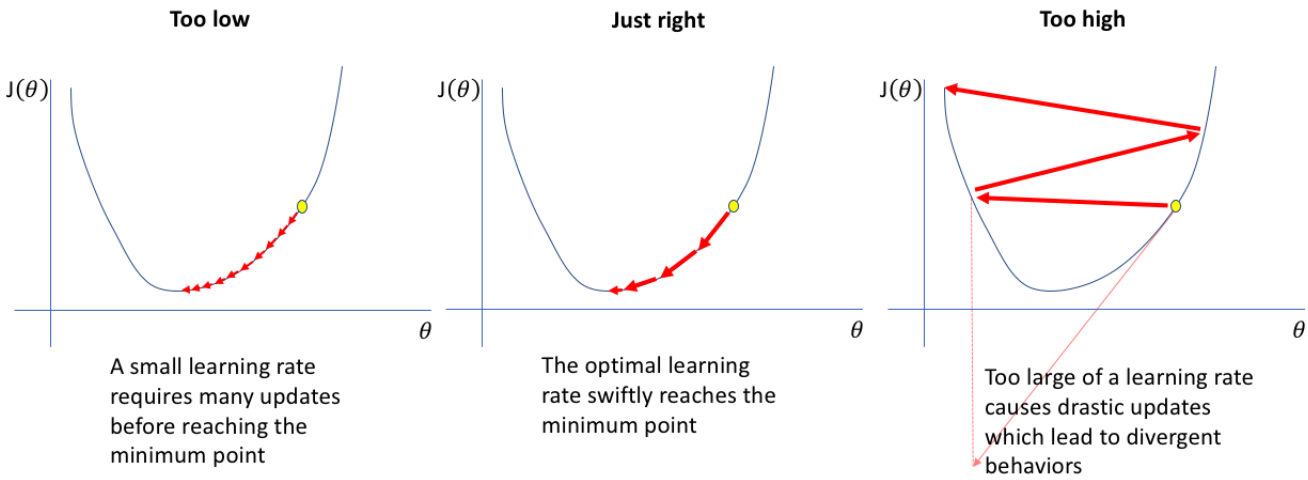


Figure 1: source = https://www.jeremyjordan.me/nn-learning-rate/

To produce the desired output with a maximized efficiency, we compare the following variables to determine the optimal parameter for simulation: time taken, convergence and accuracy. We assumed that the optimal parameter would work the same regardless of the number of processes and goods that we ran the experiment with. We decided to use ii = 10 and jj = 5 or 10 processes with 5 goods. Figure 2 and 3 show the results of our tuning. We can conclude that tau with 0.05 0.0.005 is optimal and epsilon or tolerance of $1 * 10^{(-5)}$ would produce an optimal result with a maximized efficiency.

Distance		tau							
		0.5	0.1	0.05	0.01	0.005	0.001	0.0005	0.0001
epsilon	1*10^(-4)	1.26739	0.15151	0.00010	0.00010	0.00010	0.00010	0.00010	0.00009
	5*10^(-5)	1.18197	0.14251	0.00005	0.00005	0.00005	0.00005	0.00005	0.00005
	1*10^(-5)	1.24367	0.14799	0.00001	0.00001	0.00001	0.00001	0.00001	0.00002
	9*10^(-6)	1.08250	0.12418	0.00001	0.00001	0.00001	0.00001	0.00001	0.00002
	7*10^(-6)	1.19969	0.14791	0.00001	0.00001	0.00001	0.00001	0.00001	0.00003
	5*10^(-6)	1.16474	0.18396	0.00000	0.00000	0.00000	0.00000	0.00001	0.00002
	3*10^(-6)	1.14263	0.16496	0.00000	0.00000	0.00000	0.00000	0.00000	0.00002
	1*10^(-6)	0.97920	0.16063	0.00000	0.00000	0.00000	0.00000	0.00001	0.00002
Reach Epsilon?		x	x	o	o	o	x	x	x

Figure 2: Convergence of each parameter

Time Taken(in second)		tau								
		0.5	0.1	0.05	0.01	0.005	0.001	0.0005	0.0001	Average
epsilon	1*10^(-4),	66	89	35	52	49	56	49	55	56
	5*10^(-5)	55	62	30	37	52	54	53	53	49
	1*10^(-5)	67	68	13	21	25	52	56	62	46
	9*10^(-6)	60	69	13	29	26	58	68	66	49
	7*10^(-6)	63	55	12	21	15	54	57	58	42
	5*10^(-6)	64	68	22	14	34	46	68	61	47
	3*10^(-6)	61	69	13	21	16	50	62	66	45
	1*10^(-6)	67	73	16	27	19	71	67	64	51
	Average	63	69	19	28	30	55	60	61	48

Figure 3: Time taken for each parameter

2.3 Exemplary Visualization of Euler's Method and Phi

Since it is hard to visualize the gradient descent in higher dimensions, $\mathbb{R}^m$ where $m > 3$, we ran an exemplary situation where the number of goods is 3, in other words $m = 3$. Starting from a random initial point [-.52,-.56,.64], we can confirm that the values stay on the unit sphere. After N=5000 iterations of Euler's method, the price vector converges to an equilibrium point which is a red highlighted point.

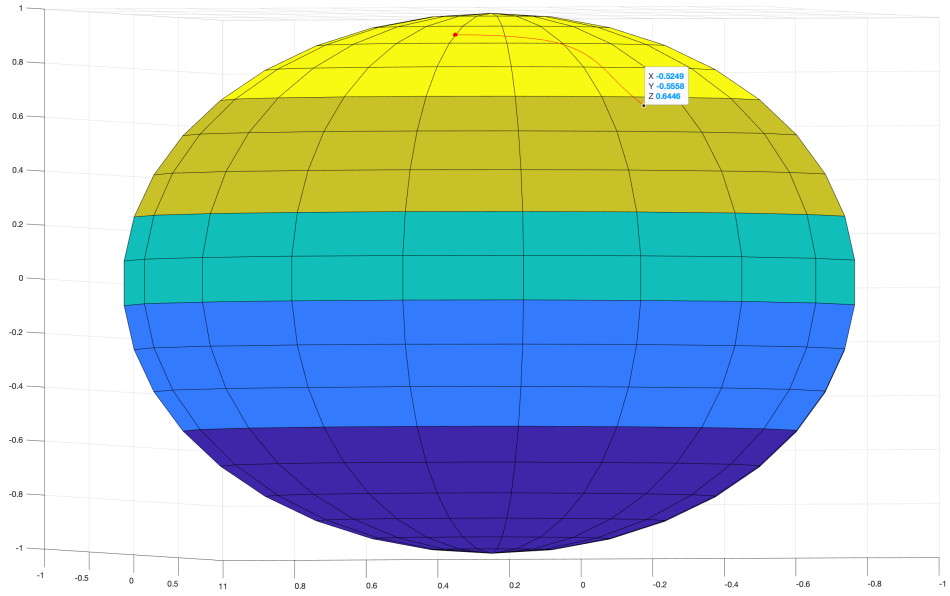


Figure 4: Euler's Method

As we can see from the graph that shows the change in phi as the number of iteration increases, a value of phi converges to 0 after approximately $170th$ iteration.

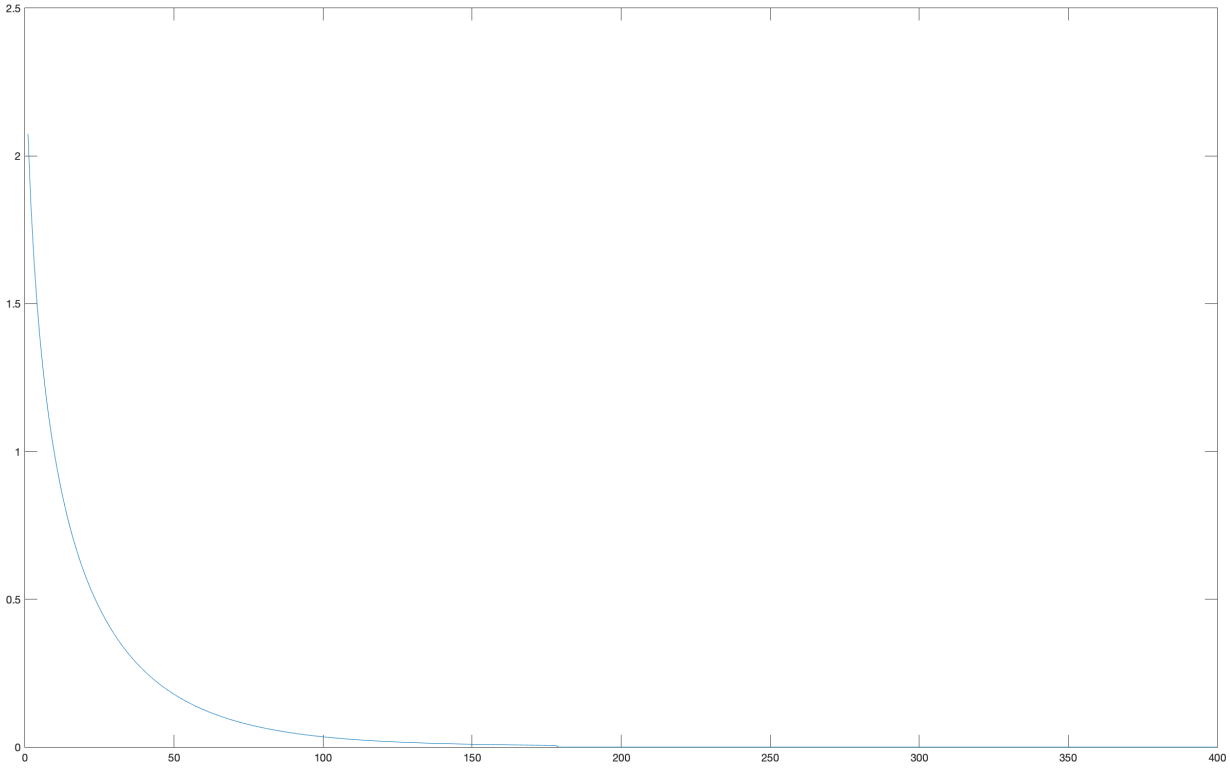
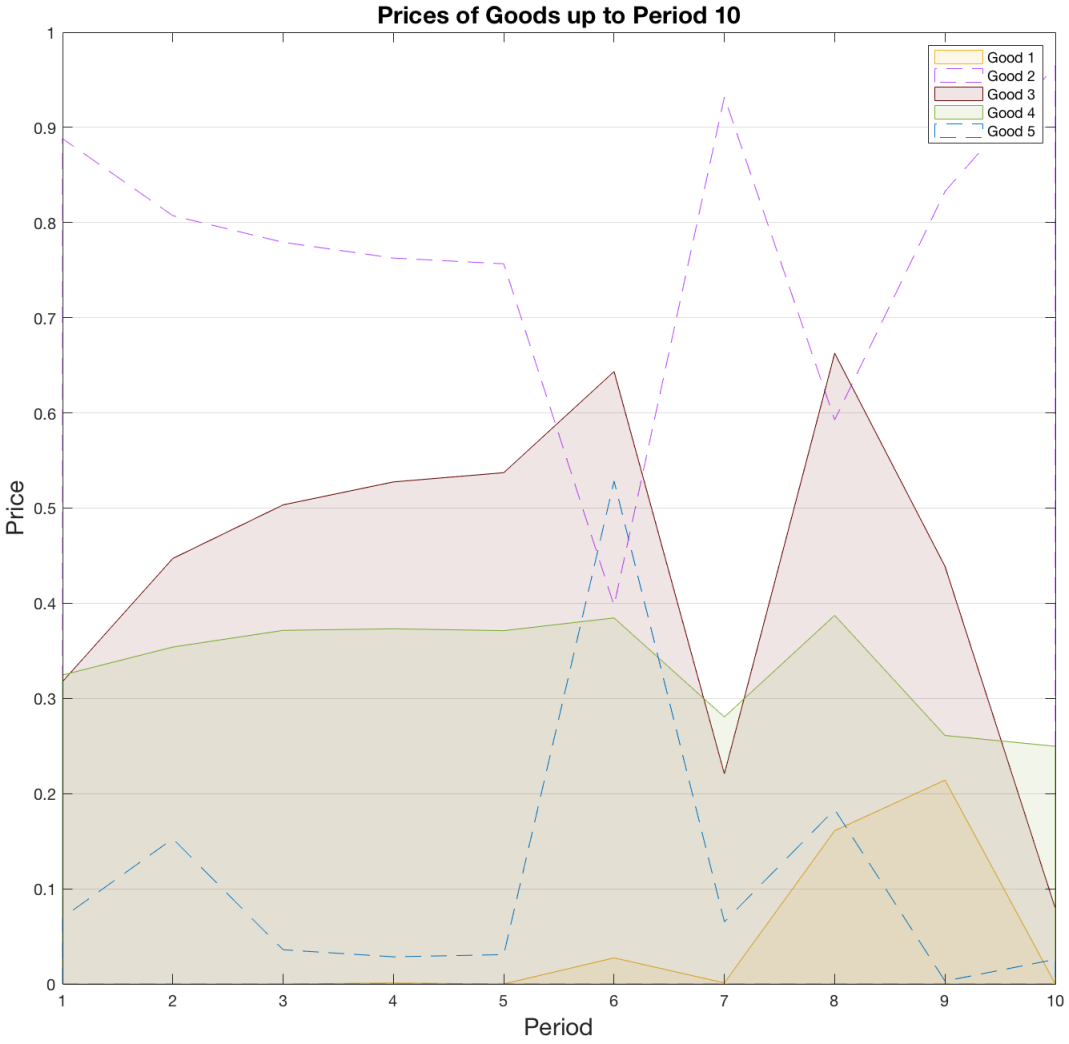
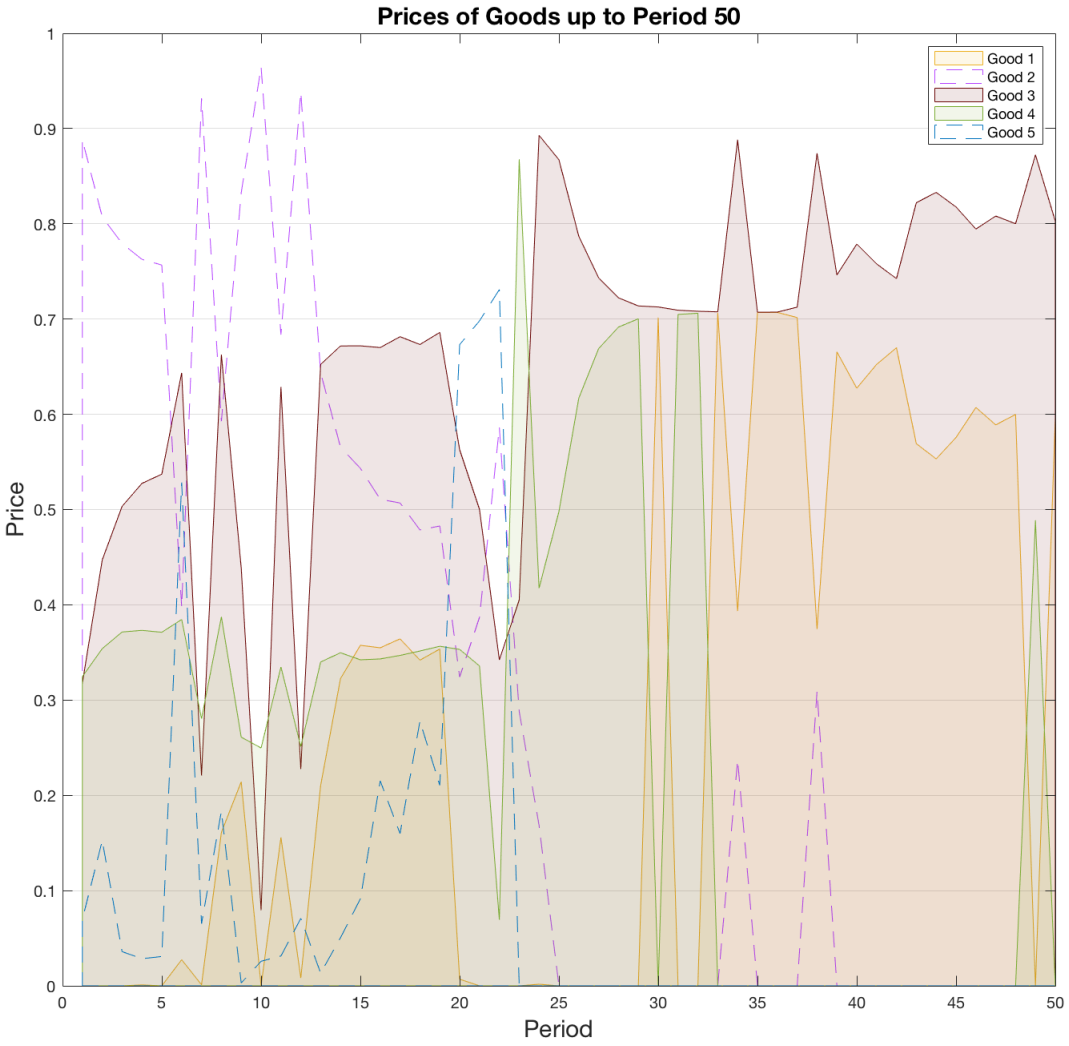


Figure 5: Change of Phi as iteration(N) increases

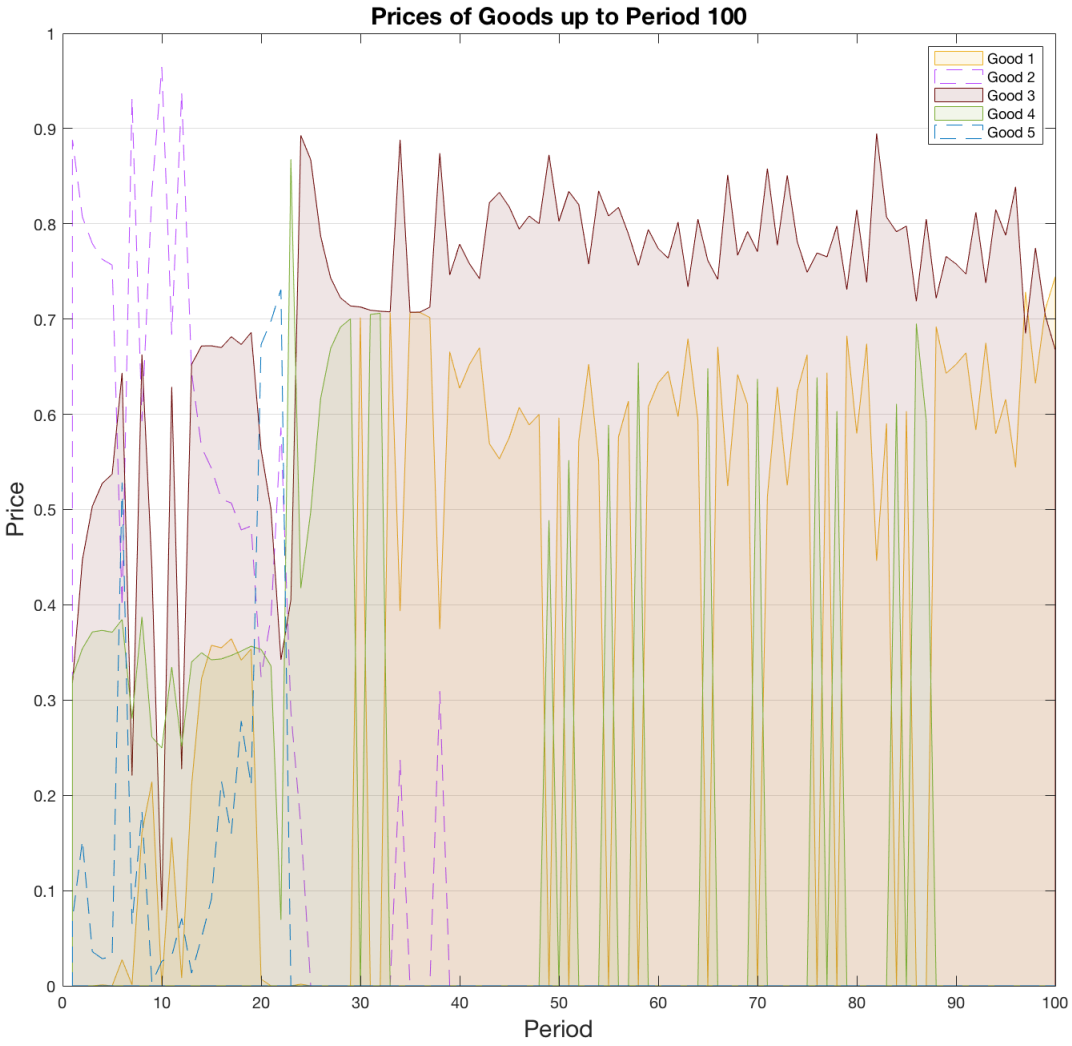
3 Results



With randomly chosen A, B matrices we found rather chaotic behavior. From periods 1-10, we see that the price of each good behaves in a nonlinear fashion. There is no single good where the price increases or decreases at a constant rate for more than two periods. Good 2 finds great initial success as well. We also see that good 1 becomes a free good.



Once we extend our analysis to period 50, we notice that our economy briefly enters a phase of balanced growth after 3 goods become free goods. Goods 3 and 4 maintain relatively high prices and remain the dominant goods for periods 23-30. As we get closer to period 50, good 4 begins to reemerge. This again shows the chaotic nature of the prices of the goods since good 4 was a free good from periods 34-49 after having great success in the beginning. Good 5 on the other hand, has not reemerged into the market since becoming a free good in period 23. It is also worth mentioning that in period 20-22, good 5 maintained a high price while good 1 slowly became a free good over the same period. Overall, good 2 and 5 slowly but surely become permanent free goods as they both do not re-enter the market past period 39.



Once we look at the simulation for 100 days, we see that good 1 and good 3 essentially become the only two goods in the market. Good 4 periodically re-emerges with a sharp climb in price only to be met with a sharp decline in price in the following period.

In order to understand why such phenomenon is observed, we studied the percentage change of price in between periods. The percentage change in price is illuminating since it shows that any big increase in price usually results in a great decline in price in the following period. This is well illustrated by the behavior exhibited by good 4 between periods 49-88. In period 54, it has a price of 0 while in period 55 the price becomes .3466. This is great increase in price and it is followed by a decline to a price of 0 in the following period. The inverse of this behavior is observed during the early periods for good 1. Good 1 is a free good for periods 1-5 and it then goes up to a price of .0274 in period 6. This is less of a sharp increase when compared to the increase from 0 to .3466 shown by good 4. In period 7, good 1 drops to .0011. Note how this drop is not as destabilizing as the drop experienced by good 4.

Our model leads us to believe that small increases in prices over time is met with less resistance than big increases in price in a short period of time. To be more specific, increases in price greater than 30% is met with a decrease in price greater than 30% in the following period.

To further test our hypothesis, we decided to simulate a monopoly. In order to make the simulation realistic, we used looked at a famous case, the United States vs Microsoft Corp. In this case, Microsoft was

accused of maintaining a monopoly position in the PC market. During the case, Microsoft had a market share of approximately 90%. Using this, we set up an A and B matrix such that one process required very little input to produce the majority of a certain good. In order to test how well these conditions held we also decided to use a large number of processes and a relatively small number of goods. We choose i=20 and j=5 or 20 processes with 5 goods. Here are those matrices:

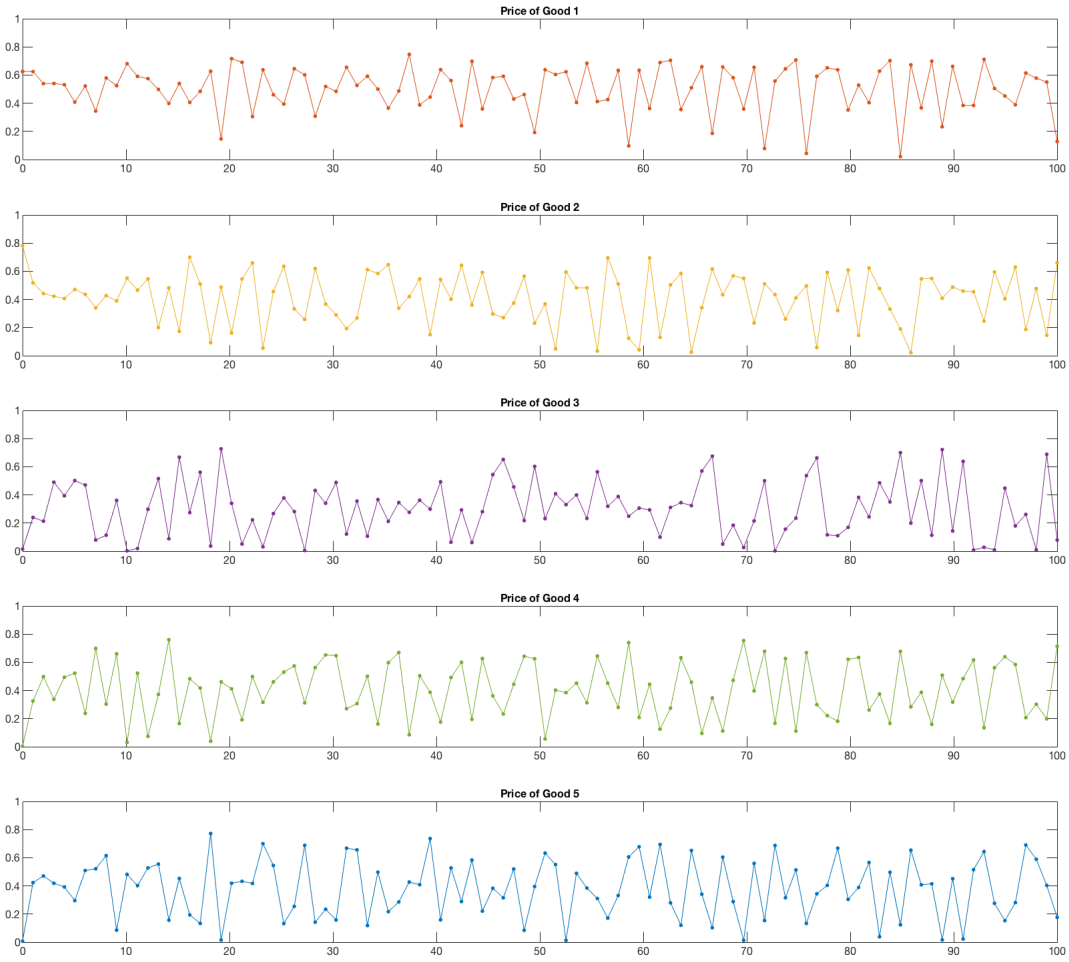
A matrix:

$$\begin{bmatrix}
\mathbf{0.22822} & 0.9718 & 0.71347 & 0.51656 & 0.96472 \\
0.67915 & \mathbf{0.12549} & 0.62999 & 0.89343 & 0.96249 \\
0.75579 & 0.78126 & \mathbf{0.19924} & 0.8424 & 0.5462 \\
0.93629 & 0.97147 & 0.5483 & \mathbf{0.16209} & 0.92294 \\
0.9547 & 0.50567 & 0.76184 & 0.82517 & \mathbf{0.21571} \\
0.50193 & 0.94029 & 0.61756 & 0.69257 & 0.62243 \\
0.60708 & 0.56774 & 0.8907 & 0.82465 & 0.82046 \\
0.94726 & 0.95103 & 0.6811 & 0.88143 & 0.65226 \\
0.7838 & 0.64482 & 0.70614 & 0.78784 & 0.91281 \\
0.81251 & 0.94321 & 0.64037 & 0.81596 & 0.94184 \\
0.69926 & 0.97297 & 0.85795 & 0.6391 & 0.97269 \\
0.94728 & 0.56451 & 0.81115 & 0.91992 & 0.6954 \\
0.62751 & 0.84831 & 0.64591 & 0.71342 & 0.90066 \\
0.95452 & 0.57491 & 0.78147 & 0.81581 & 0.57856 \\
0.94223 & 0.83853 & 0.82898 & 0.91673 & 0.81258 \\
0.69162 & 0.8918 & 0.61538 & 0.63509 & 0.84949 \\
0.95481 & 0.74978 & 0.57618 & 0.7004 & 0.54293 \\
0.76117 & 0.91983 & 0.74777 & 0.77713 & 0.76559 \\
0.9879 & 0.61739 & 0.63183 & 0.72193 & 0.94428 \\
0.51863 & 0.71487 & 0.51631 & 0.54519 & 0.87219
\end{bmatrix}$$

B matrix:

$$\begin{bmatrix}
\mathbf{0.93588} & 0.15726 & 0.22859 & 0.10434 & 0.19759 \\
0.14041 & \mathbf{0.99536} & 0.23516 & 0.27168 & 0.33761 \\
0.22592 & 0.25976 & \mathbf{0.95045} & 0.37771 & 0.36972 \\
0.26345 & 0.37034 & 0.11555 & \mathbf{0.96541} & 0.34258 \\
0.20047 & 0.1686 & 0.34672 & 0.39085 & \mathbf{0.95629} \\
0.14964 & 0.10844 & 0.38661 & 0.1802 & 0.35817 \\
0.20114 & 0.2753 & 0.24069 & 0.30578 & 0.38173 \\
0.32357 & 0.37198 & 0.30401 & 0.34862 & 0.30406 \\
0.34294 & 0.1003 & 0.25449 & 0.15513 & 0.37523 \\
0.32381 & 0.16477 & 0.25662 & 0.18998 & 0.17701 \\
0.13503 & 0.37757 & 0.23883 & 0.22336 & 0.36569 \\
0.31986 & 0.2567 & 0.18885 & 0.17095 & 0.37601 \\
0.17745 & 0.15416 & 0.31161 & 0.15852 & 0.19002 \\
0.24867 & 0.19034 & 0.32007 & 0.37568 & 0.12202 \\
0.14609 & 0.10867 & 0.10273 & 0.27894 & 0.28271 \\
0.15675 & 0.13926 & 0.35255 & 0.36847 & 0.12549 \\
0.26643 & 0.36378 & 0.26736 & 0.3257 & 0.31863 \\
0.16204 & 0.30829 & 0.38511 & 0.18843 & 0.23437 \\
0.3659 & 0.13421 & 0.23276 & 0.29786 & 0.29537 \\
0.24111 & 0.10423 & 0.29014 & 0.25943 & 0.15085
\end{bmatrix}$$

The bold numbers indicate the parameters that give the advantages described above.



In a monopoly economy, the firms are price setters. They can essentially charge as a high of a price that they can get away with. However, if the price is too high, the demand will begin to decrease forcing the price to go down. This phenomenon is is clearly observed in our results. In period 14, the price of good 2 is .0398. It then increases to .2320 in period 15 and then decreases to .0298 in period 16. One interesting feature of the monopoly simulation that is that there are no free goods. Goods will remain in production regardless of how low the price becomes. This is also consistent with features of a monopoly market since such markets have goods with no close substitutes.

4 Further Study

Our paper lacks formal mathematical reasoning in why such behavior is observed. There might be an analytical method that determines exactly how much a certain increase in price will affect future prices.

For further study, we suggest to simulate with a complicated real world setting. One can have different A and B matrix for each time period for each good. For example, one can utilize the production and consumption data

5 Acknowledgment

This project would not be possible without Professor Charles Peskin's notes on the Von-Neumann model and economic equilibrium which can be found here. Special thanks to Guanhua Sun for teaching us how to use MATLAB effectively. All our code is available at our GitHub link. Suggestions are welcomed.