Independent Research Report
Learning about Eisenberg-Noe Clearing Vector

Wilson(Wei) Wang

Abstract

In this independent research, I follow the intuition of Professor Feinstein and run Eisenberg-Noe clearing vector in different models and demonstrate them in graphic form. In fixed-time model, given the interbank liability matrix and bank cash injection vector, I calculate the clearing vector p_0^* for a bank system at a given time t_0. In dynamic discrete model, I use overtime interbank liability and cash injection to generate overtime bank cash flow and clearing vectors. I keep track of bank system activities over time period n, and obtain clearing vectors $p_1^* \sim p_n^*$ at time $t_1 \sim t_n$. Furthermore, I execute the model in larger datasets. To simulate multiple characteristics of a typical bank clearing system, I generate the input data based on several random graph models, then apply the Eisenberg-Noe model. Last but not least, adding fixed amount of shocks to different bank systems, I then compare the behavior of the models horizontally. The comparison helps me understand the effect of different structural characteristics on the stability of a bank clearing system.

Applying the methodology to different models, I learn about using Eisenberg-Noe vector to capture defaulting banks, and about how bank defaults have cyclic impact on the bank system and can cause future rounds of defaults. Furthermore, given different structures of bank systems, Eisenberg-Noe method does well to explain how the bank default spreads across the system and how tightly future rounds of default depend on one or multiple banks in the system. During the study, I also learn the dynamic discrete Eisenberg-Noe model is a relaxed version of its fixed-time counterpart, it allows banks to temporarily "default" and generate cash flow. I plot the cash flow and clearing vectors over time for a typical bank recovered from temporary default.

1 Introduction

1.1 First Dataset: A toy model of five banks

To understand about a financial system, I start from a simplified model of five banks. In this model, each bank has some initial endowment and obligation to other banks. Assuming that when a bank is unable to make its payments in full, it does so in proportion to what it owes. Knowing which banks are default in a certain round, I then apply Eisenberg's method and calculate the clearing vector. The form below gives me the information of the future transaction between different banks as well as their expected future cash infusion.

From\To	Bank0	Bank1	Bank2	Bank3	Bank4
Bank0	0	1	4	3	1
Bank1	2	0	3	0	2
Bank2	0	3	0	4	0
Bank3	1	2	3	0	2
Bank4	4	4	2	1	0
Cash Infusion	3	3	5	2	7

1.2 Eisenberg-Noe Clearing Vector at a Fixed Time Spot

Given the interbank transaction matrix and cash injection vector, I am able to calculate the clearing vector based on Eisenberg's method because the model satisfies three criteria below:

- Limited liability: Total payment made by a bank is less or equal to its cash flow available.
- Priority of debt claims: All outstanding liabilities are paid off first before stockholders receive any value.
- Proportionality: If default occurs, all debt holders are paid in proportion to the size of their original claim.

To perform the Eisenberg method, we follow the steps below:

1. Denote the original liability matrix as L.
2. Calculate the total liability vector, where each node $\bar{p}_i = \sum_{j=1}^{n} L_{ij}$ denotes the total liability the corresponding bank owes.
3. Calculate the relative liability matrix, where each node $\Pi_{ij} = \begin{cases} \frac{L_{ij}}{\bar{p}_i} & if\ \bar{p}_i > 0 \\ 0 & otherwise \end{cases}$ denotes the proportion of wealth $Bank\ i$ should pay to $Bank\ j$ out of its total payment $\bar{p}_i$.
4. Calculate the total cash flow each bank will receive in order to pay off its debt, as: $\sum_{j=1}^{n} \Pi_{ij}^{T} p_j + e_i$
5. Therefore, in this iteration, the clearing vector would be the amount each bank will actually pay, and an approximately calculation would be:
   $\min(total\ amount\ owed, total\ amount\ received)$.
   However, the result from step 4 only gives us an overestimated value of total amount received, because it assumes other companies will pay in full (which does not always

happen because they could default), therefore we will treat this clearing vector as the new total obligation and plug it bank into step 4 to calculate the new total amount received. Fortunately, Eisenberg has proved mathematically in his paper that this process is going to converge in no more than n steps, n is the total number of banks.

6. The iteration above is essentially a mapping function, in which each iteration maps the total obligation to an updated clearing vector (which gets closer to the exact clearing vector as we iterate). The mapping function can be addressed as:
$$FF_{p'}(p) = \Lambda(p')\big(\Pi^T\big(\Lambda(p')p + \big(1 - \Lambda(p')\big)\bar{p}\big) + e\big) + (1 - \Lambda(p'))\bar{p}$$
where $\Lambda(p)_{ij} = \begin{cases} 1\ i = j\ and\ bank\ i\ is\ defaulting \\ 0\ otherwise \end{cases}$,
and p' here is a "super-solution", which indicates the defaulting companies in current round. Running this iteration for n times, and we get the clearing vector, which provides the information how much each bank ends up paying and whether it defaults or not. In his paper, Eisenberg has proved that the clearing vector converges in n iterations, and the solution is unique given we start our iteration from a super solution p'.

1.3 Eisenberg-Noe Vectors Overtime: The discrete dynamic method

In the discrete dynamic model, we represent the future transactions among firms in a three-dimensional matrix, which can be viewed as a collection of transaction matrices over time. Our cash infusion vector now becomes a matrix where the $t_0\ th$ column corresponds to cash infusion at time t_0.

Notice that we could run the Eisenberg model over time. To start with, we run the fixed-time model at time 0, and obtain a unique solution. For each time period t, we have a deterministic clearing vector as super solution (clearing vector at time $t - 1$ plus total income at time t). To avoid heavy computation in our program, we use the total debt, a relaxed super solution, to initiate the iteration.

This model allows that a defaulted bank to continue operation and recover its debts through future cash injection and transactions. This approach is more realistic than the fixed-time model because the default happens in a time span instead of a time spot and the bank would only truly "default" if it fails to recover its debts at the time n. Given that transaction and cash injection information is public, the debtors could choose to lay off the debts of a company if the company will recover from default and pay their debts in full.

Based on this intuition, a bank in our model has one of these states:

1. Solvent: The bank succeeds to pay all the outstanding debts.
2. Default: The bank fails to pay all the outstanding debts, and all its future cash inflow will be distributed to debtors until all debts been paid.
3. Out of system: The bank fails to pay all the outstanding debts at the end of time span, and is removed from the bank system.

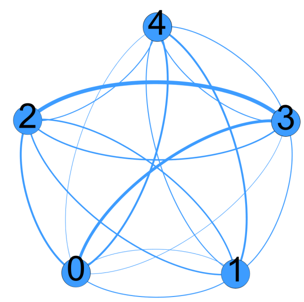
Since we base our model on Eisenberg's method, we will have a unique solution at any time. It can be obtained given the transaction and cash injection information. Given the clearing vector at time $t-1$, we can calculate the debt payment vector at time t.

The revived banks generate a cash flow similar to the defaulting banks: distribute all their incoming wealth to all debt holders in proportion to the amount they owe. Once they have paid all debts they owe, they start to behave like a normal bank, as they pay the exact amount they owe.

Given the above facts, we are able to treat the system as a continuous time model. We use dynamic discrete model to keep track of banks status over time.

1.4 Graphical representation

The data above can be represented in a single directional graph. Each node represents a company whereas each edge stands for the net transaction between two banks (the transactions between banks are usually bidirectional, and cash flow direction is read as clockwise). Each edge width stands for the amount of cash flow between its connected entities.



Each node represents a certain bank, and each arrow represents the direction of cash flow. For example, the arrow pointed from bank 4 to bank 2 means that at the end of this time period bank 4 will pay the debt it owes to bank 2. The width of each arrow represents the amount of money the banks are paying each other. Wider arrows represent a larger amount of cash flow.

Observe the graph, we can see that bank 0 and bank 4 are both in debt to several other banks. Increasing the strength of these edges (thus increase the debts) of those banks may cause them to default which may or may not result in a financial distress.

2 Several categories of shocks we can apply to the clearing system

2.1 Decrease cash infusion

The cash infusion to a certain bank may change based on its day-to-day operation and client credibility. When the change in cash infusion is too small to cause a bank default, its impact will limit to change the cash position (after getting back its loans and paying its debts) of the particular bank and have no impact on other banks in the system. However, when the situation is severe enough, the bank will default which may or may not cause a second round of default or a financial distress across the system.

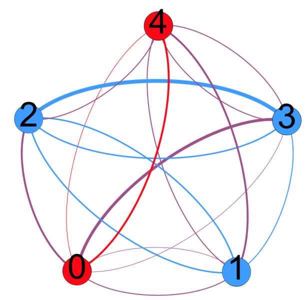
2.2 Change transaction claim

Unlike the change in cash infusion, the change in interbank transaction claims happen when Bank A have to pay bank B an amount that is different from the original claim. It is a zero-sum game within the bank system, so it doesn't change the total wealth in the system directly. However, when a bank has to pay more or receive less, there is a chance that it will default and cause future rounds of default or financial distress.

2.3 Applying the shocks to toy bank systems

To analyze the robustness and resilience of our model, we apply different kind and level of shocks to the system and see how it holds. First, we will explore the situation where nonsystematic shocks are applied and see how susceptible is our bank system to the default of a particular bank. Furthermore, we will test the effect of shocks on the whole bank system.

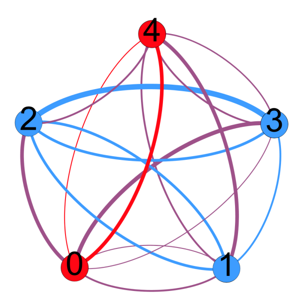
We now illustrate two typical shocks in our toy model and plot the result.



In our first shock, we cut the cash infusion for all the banks to a half of original. This simulates a systemic impact to income of the whole bank system.

This impact results in the default of bank 0 and bank 5 (indicated as red node in the graph). A good explanation for the default of these two nodes is that they rely on their future cash infusion to pay a large proportion of their outstanding debts, so that they are affected very negatively by a cut in cash injection.

In our second shock, we double the amount banks owe to bank 3.

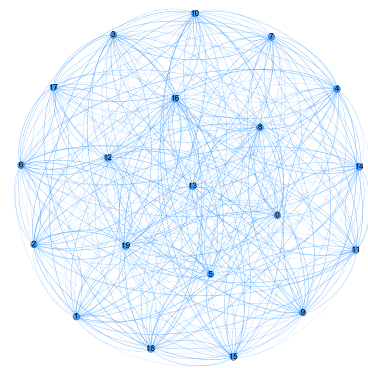


Different from the last shock, yet yielding very interesting result. It seems that the change of claims does not change the whole wealth in the bank system, but it causes nodes to meet their constraints (each company should have zero or more cash remaining at any given time period).

3 A variety of toy bank models

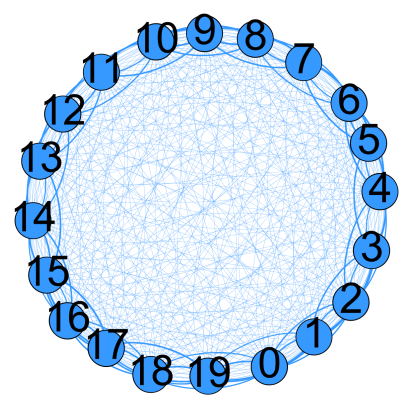
3.1 Random Graph

In random graph model, I make the assumption that each bank in the system has approximately the same amount of transaction with every other bank. Each cash flow follows a uniform distribution, and the cash injection data is generated based on the transaction matrix to ensure no bank defaults when the system is stable. This is intuitive because banks are likely to match their transactions to their cash injection, so that they will keep a fixed percentage of cash in reserve and the other in a variety of claims. When their claims default, they will use the reserved cash as a buffer. In the graph on the right-hand side.



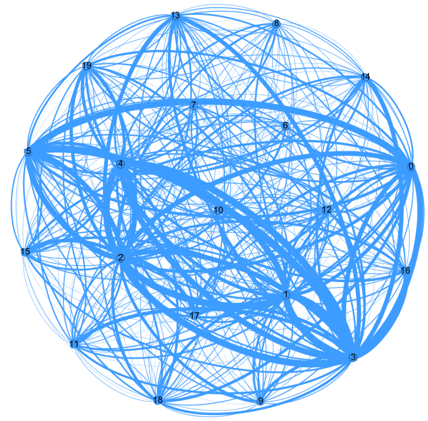
3.2 Watts–Strogatz

In the Watts–Strogatz model, each bank has higher value transactions with its neighbors. This is to simulate the fact that local banks may have more transaction and claims with each other. To construct this data, we started from a ring lattice indicating the network of banks having strong relationship with each other. Not excluding the transaction between a bank and random banks, we allow transactions between all banks with a reduced average cash flow volume.



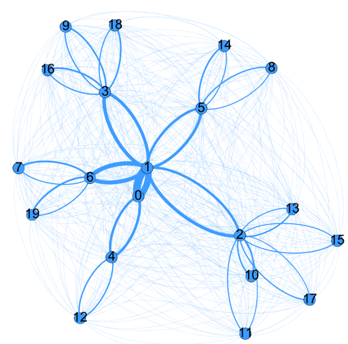
3.3 Barabási–Albert Model

In the Barabási–Albert model, we introduce parts of the bank into the system and generate their transaction matrix. We assume that those banks are the "old banks", and we gradually introduce new banks into the system. Since "old banks" are likely to be well-known and considered more reliable, each new banks introduced into the system is more likely to have more business to do with the "older banks". In the graph, we observe that some "old banks" have much higher transaction volumes than other banks.



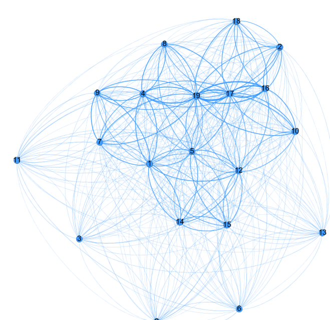
3.4 Tree graph

In the tree graph, we observe the phenomenon where central banks have transactions with each other whereas their branches and regional banks have much less transaction on average, and they seldom have large claims with a remote bank. On the other hand, regional banks are likely to have regular transactions with the banks they are associated with. A tree graph represents this leveled structure.



3.5 Random Unions

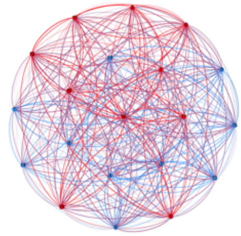
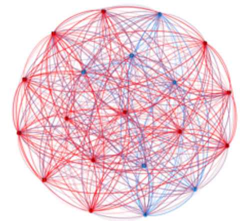
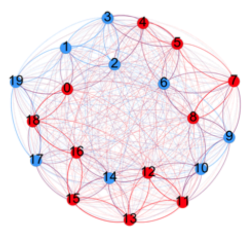
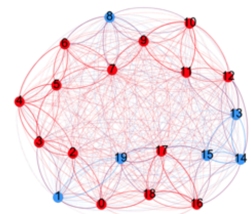
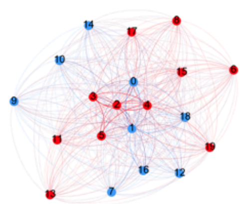
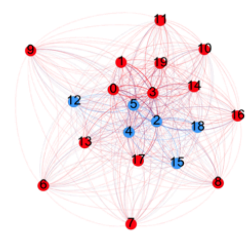
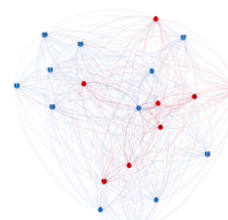
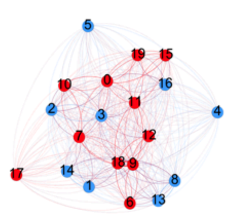
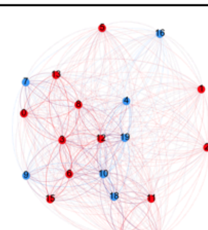
The random union model studies more about a regional bank system, where banks are close to each other and can form unions. There are small councils of banks where banks in the same council can join a plan and have some transactions. In this model, each bank is allowed to join as many councils as it likes and each council provides the bank with some transactions. On the other hand, a bank can also join no council, so that it will mostly rely on its own cash flow to operate. In this model, we assume a fixed council-to-bank ratio, and a fixed council size (about 1/5 of the total number of banks in the system).



4 Eisenberg-Noe solution for toy bank systems under shocks

Although there are a variety of toy models, the Eisenberg-Noe fixed time model represents them in a transaction matrix and cash injection vector. Applying different financial shocks to those banks systems and calculate clearing vectors based on Eisenberg-Noe method, I obtain some results that are quantitative and comparable from model to model.

4.1 Eisenberg-Noe clearing vector under reduced cash injection

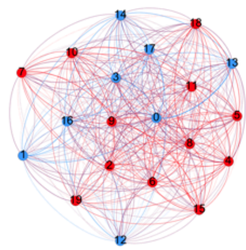
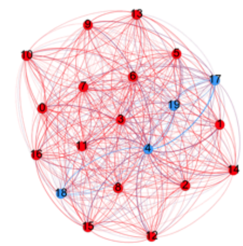
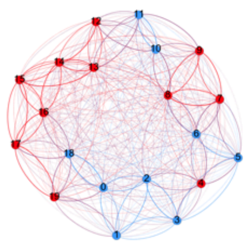
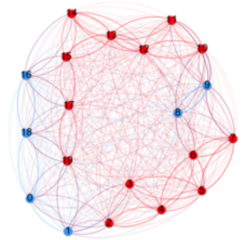
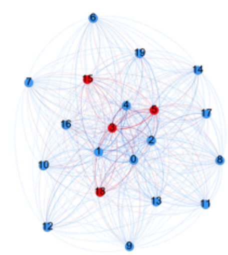
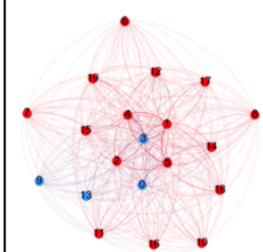
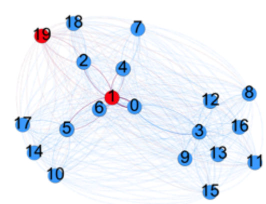
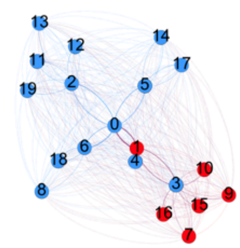
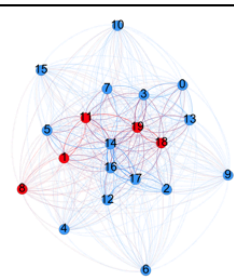
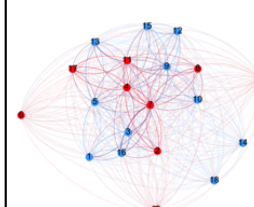
Method	default ratio lower bound	graph lower bound	default ratio upper bound	graph higher bound
random graph	50%		65%	
watts strogatz	55%		70%	
barabasi albert	55%		70%	
tree graph	15%		40%	
random union	55%		65%	

In the graph above, I set up stable models where no bank default before the shock, and then I cut the cash injection to each bank in the system to a half. The random graph gives me a benchmark of $50\%{\sim}65\%$ as our benchmark, whereas watts strogatz, barabasi albert graphs, and random union model yield slightly higher default ratio. Whereas the tree graph appears to be much more immune the systematic shock in cash injection for some reason. It has a much smaller default ratio range of $15\%{\sim}40\%$.

Observing the graph, we can see that the graph models reach their bound only under some specific conditions.

- In Watts Strogatz's model, most banks default because they are effected by defaulting neighbors. A chain of defaulting neighbors cause a "cloud" of red dots, where most bank in that neighhood default and they negatively impact each other's cash flow. On the other hand, in the scenarios where defaulting ratio reaches its lower bound, we can see that the defaulting banks are spread out almost evenly, so each bank in the system does not have too many defaulting neighbors and some of them manage to survive.
- In Barabasi Albert model, we observe some banks with every tight transaction with other banks. Under the scenario where most banks default in the shock, most of the large banks survive, and this is most likely that they have a well-balanced and good amount of cash flow from claims with smaller banks. Obviously, this zero-sum game does not favor the smaller banks, so that most of them default under this scenario. In the case where only 55% banks default, the tide turns and most of the larger banks default whereas the smaller banks survive. We manage to catch these extreme cases in our model due to the randomness of our data and the number of time we run the experiment.
- The tree graph is a unique model, because it provides a super stable bank structure that behave even better than the random graph. Analyzing the graph, we can understand that since most banks are associated with only their superior and subordinates, they are not effected by cross-system shocks that much. This allows the scenario where severe defaults happen in one part of the system while the other parts of the system remain stable. Under the scenario where 15% banks default, only the central banks default and the regional banks remain intact.
- The random union model provides a super stable result, where its spread between lower bound and upper bound is 10% which is even smaller than that of the random graph model. This indicates that under random union model, the number of defaulting banks is almost fixed. The intuition is that while banks in the unions behave somewhat like in the random graph model, banks out of the system rely almost entirely on their cash injection and are not effected much by the defaulting of banks in the unions. This factor adds a second level of randomness which gives the system a result that is closer to the average (parts of the banks follow the random graph model, and the rest banks are drawn from a random defaulting bank sample).

4.2 Eisenberg-Noe clearing vector when changing transaction claims

Method	default ratio lower bound	graph lower bound	default ratio upper bound	graph higher bound
random graph	60%		80%	
watts strogatz	55%		70%	
barabasi albert	20%		80%	
tree graph	10%		30%	
random union	25%		45%	

The default ratio lower bound and upper bound behave differently under this scenario. When we reduce the cash injection, the shock is applied to every bank almost evenly. Now we change the claims of a certain bank x, the shock impacts only banks that are tightly connected to x. The scale of bank default under this shock therefore can be explained qualitatively with the war the models connect the banks.

Observing the graph, we see the phenomenon below:

- Random graph gives us a much higher default ratio, the higher bound is higher than any other bank models and so does its average default ratio. This is due to the uniform distribution of connections in the random graph. When a bank creates strong negative cash flow for its neighbors, it reaches out to almost everyone in the bank system and this shock from an individual therefore applies to the whole bank system, giving all banks in the system a significantly higher chance to default.
- Watts Strogatz model shows that the defaulting banks are almost all connected firmly to each other. This follows the Eisenberg-Noe model well, because in the Eisenberg-Noe model, we assume banks default in a sequence that they will impact tightly connected banks if they default and cause rounds and rounds of bank defaults. Depending on the severity of the shock, we have different amount of banks default. Once a defaulting bank fails to pass on the default to its neighbors, the chain reaction terminates and the rest of the banks stay solvent.
- Barabasi Albert shows us a very widely spread range of default ratio from $20\%\sim80\%$. The lower bound happens when some insignificant small banks default, and the negative cash flow caused by their default is too small to damage the bank system. On the other hand, when a large bank who has many transactions get effected by the shock, the huge amount of negative cash flow quickly spread out across the system and cause most banks to default (similar to the impact in a random graph).
- The tree graph stays relatively stable given that the default of a bank can only have effect on its neighbors. In the best-case scenario, the default of a bank fails to spread to its neighbors and therefore almost no bank in the system gets effected by that one bank's default. In the worst-case scenario, almost all banks in a whole branch of the bank tree system default, but their default has virtually no impact on banks under other branches.
- In the random union, when the defaulting ratio is low, the bank default only exists in some of the unions that get involved with the bank who creates the negative cash flow. Banks out of the unions are generally safe because they don't have much transactions with the problematic bank and its neighbors. Under the situation where defaulting ratio is higher, the default manages to spread to a few more banks, but since out-of-union banks are not tightly connected the default fails to spread across the whole bank system.

4.4 Bank cash flow and clearing vector overtime under Eisenberg-Noe model

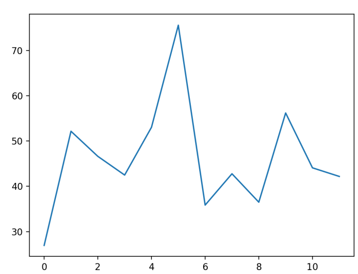
Using random graph model as the toy banks system, and reduce the cash injection for all 10 banks in the system. We can see their default status overtime. In the status matrix, “0” means solvent and “1” means default. In our model, 5 of the 10 banks default at the end of the time period.

```
[[0 0 0 0 1 0 0 1 1 1 1 1]
 [1 1 0 0 0 0 0 1 0 0 0 0]
 [1 1 0 0 1 0 1 0 0 0 0 0]
 [1 1 1 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 0 1 1 1 1 1 1 1]
 [1 1 0 1 1 0 0 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 0]
 [1 1 0 0 0 0 0 1 1 0 0 1]
 [0 0 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 1 1]]
```

As we can see, almost all the banks in the system have experienced stages of temporarily default, but since their debts get rolled over to later periods, they manage to pay off their debts and remain solvent.

Since I made the random graph toy model assuming banks only have a very small proportion of cash reserved after paying off their debts, a reduction in cash rejection causes severe defaults in the bank system overtime. In reality, banks manage their risk well, and the defaulting patterns in the matrix will only apply to a very small number of banks or not at all.

The clearing vector overtime for a random bank in the toy model looks like the graph on the right: The clearing vector overtime fluctuates a lot because if the bank default, all its cash flow generated will be used to pay off the debt. When the cash flow is low, the banks only pays a very small amount of cash to the debt holders. On the other hand, when the bank generates a large amount of cash, it has to pay all the money it owes which result a huge clearing vector for that time period.



Just like in 4.1 and 4.2, we are able to plot defaulting banks overtime in graph model and for each bank system, we can generate defaulting banks overtime. Eisenberg-Noe model yields one and only one solution for the clearing vectors overtime, and it proves to be a powerful tool to study about bank cash flows. With realistic data, quantitative sensitivity analysis, and generalization, Eisenberg-Noe model can generate valuable results for risk management by the professionals.