Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

Contagion Risk in Interconnected Banking Ecosystems

Arno Bargerbos   Erasmus Elsner   Dorela Kozmai

Zurich
Dec 2014

Agreement for free-download

Arno Bargerbos Erasmus Elsner Dorela Kozmai

Declaration of originality

This signed "Declaration of originality" is a required component of any written work (including any electronic version) submitted by a student during the course of studies in Environmental Sciences. For Bachelor and Master theses, a copy of this form is to be attached to the request for diploma.

I hereby declare that this written work is original work which I alone have authored and written in my own words, with the exclusion of proposed corrections.

Title of the work

Contagion Risk in Interconnected Banking Ecosystems

Author(s)

Last name	First Name	
Bargerbos	**Arno**	
Elsner	**Erasmus**	
Kozmai	**Dorela**	[illegible]

With my signature, I hereby declare:

- I have adhered to all rules outlined in the form on „Citation etiquette", www.ethz.ch/students/exams/plagiarism_s_en.pdf.
- I have truthfully documented all methods, data and operational procedures.
- I have not manipulated any data.
- I have identified all persons who have substantially supported me in my work in the acknowledgements.
- I understand the rules specified above.

I understand that the above written work may be tested electronically for plagiarism.

Zurich, 12. December 2014

Place, Date

Erasmus Elsner
Dorela Kozmai [illegible]
Arno Bargerbos

Signature*

* The signatures of all authors are required for work submitted as a group. The authors assert the authenticity of all contents of the written work submitted with their signatures.

Contents

1 Abstract

The recent banking crisis has made it evident to regulators that the importance of a bank within the financial system cannot be determined in isolation, i.e. based on capital adequacy alone. In particular, the crisis has shown that the position of a bank within the interbank network can matter significantly. If a bank is highly interconnected by way of interbank lending and borrowing, its failure will more likely result in shock propagation throughout the banking ecosystem.

Following Nier et al. [9] we model an interbank credit network as an $N \times N$ adjacency matrix and analyze the effects of different network structures on shock contagion. Building on the MATLAB code of Kalinowski Pawel, Park Sanggil, Walser Dario from this spring semester's course, we extend the analysis of shock contagion to more sophisticated network topologies. Firstly, in line with empirical evidence, which suggests power-law distributed interbank network structures, we randomly create scale free interbank networks using a preferential attachment algorithm and then analyze shock spreading within such a setting. Furthermore, we randomize the interbank borrowings such that not every loan has an equal value. Finally, we respond to calls by policymakers and prudential banking supervisors, by introducing a basic regulator that focuses and regulates the interconnectedness of banks.

2 Individual contributions

All team members participated in regular team meetings and regularly exchanged code, results and ideas over email. The final report was written by all team members.

3 Introduction and Motivations

The global financial crisis of 2007-2008 has exposed major weaknesses in the area of global financial supervision. In particular, it has been criticized that financial supervision to this point has been mainly confined to micro-prudential regulation. Thereby the focus has solely been on the capital adequacy of individual institutions, while missing out on the larger picture – in particular ignoring the level of interconnectedness of financial institutions in the system and the associated contagion risks.

As a result, there has been a consensus call from policymakers, prudential regulators and systemic risk surveillance bodies, in particular the Bank for International Settlement (BIS), the Financial Stability Board (FSB), and the International Monetary Fund (IMF), to establish methods for analyzing and regulating banks that are too-interconnected-to-fail (TITF) [3].

At the core of the too-interconnected-to-fail (TITF) problem is the risk that the default of one institution can lead to successive rounds of failures by other banks in the ecosystem. This default cascade is a direct result of the inter-bank credit exposures in place and the interbank market network topology. As a result, the macroeconomic importance of both analyzing the topology of the interbank network and regulating banks based (at least partially) on some measure of inter-connectivity criteria appears necessary. It is for this reason that we are following up on last semesters project, in which Kalinowski Pawel, Park Sanggil and Walser Dario [1] investigated the stability of interconnected banking networks based on the work of Nier et al. [9] By extending their analysis of banking networks to to more sophisticated models, we hope to answer:

- What happens in the banking ecosystem of Nier et al. [9] where the interbank loans are mapped on a scale free network with power-law distributed degrees, meaning that few banks are highly interconnected while others operate on the periphery of the interbank market?
- What is the impact to systemic risk, proxied by the number of defaults, if a regulator is introduced who focuses and regulates some measure of interconnectedness of banks?
- How well does a regulator that randomly changes interconnectedness do compared to a regulator that does so in a focused manner and how does he do in either a (i) random graph network and (ii) a scale-free network?

4 Description and Implementation of the Model

Following Nier et al. [9], we model a banking system as a network of nodes $N = \{1, ...n\}$, with each node representing a bank and each edge representing a directional interbank lending relation between two banks (see also [8] and [10]). The interbank network is thus represented by a $N \times N$ adjacency matix:

$$x = [x_{ij}]_{i,j} \in N$$

$$x_{ij} \in \{0, 1\}$$

Thereby x_{ij} equates to 1 if bank i has made an interbank loan to bank j and 0 if no such interbank loan has been put in place:

$$\begin{bmatrix} 0 & \cdots & x_{1j} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i1} & \cdots & 0 & \cdots & x_{iN} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nj} & \cdots & 0 \end{bmatrix}$$

The zeros along the diagonal of the matrix indicate that the banks cannot make any interbank loans to themselves - intra-group loans within the banks are therefore disregarded.

4.1 Generating the Balance Sheet

While the particular topologies of the networks utilized in our project are described in section 4.2, we can construct the individual banks' balance sheets (consistent with Nier et al. [9]) from this basic setup of the interbank lending network. The function that goes through this process, $generate_banks.m$, was written by the previous group that worked on this project [1], but we nevertheless provide a brief description as it forms the basis of our simulations. In a first step, the total external assets, denoted by E, are set exogenously. Notably, E represents the aggregate loan volume of the entire banking system to ultimate retail and corporate borrowers outside of the banking system. As such, this represents the net funds provided by the banking system to the real economy.

In a second step, the ratio of aggregate interbank loans (I) to total assets (A) in the banking ecosystem , denoted by $\theta = I/A$, and thereby of external assets to total asset, denoted as $\beta = E/A$ are set.

The initially generated interbank network has directed edges with binary values of 0 or 1. In a third step, a uniform loan value w is established by dividing the aggregate interbank loans I by the total number of directed edges, denoted as Z, i.e. $w = I/Z$. This principal amount of the individual interbank loans is then applied to the adjacency matrix, which enables us to generate vectors setting out the banks' interbank loan assets i_i and interbank loan liabilities b_i.

We first define the interbank assets i, the sum over j columns:

$$i_i = w \sum_{j}^{N} x_{ij}$$

Further to lending to other banks through the interbank loan market, each bank also lends to retail and corporate borrowers outside the banking system. This outside lending is represented by external assets denoted by e. The individual bank's assets, denoted by a, are therefore calculated as the sum of external assets e and interbank assets i, i.e. $a_i = e_i + i_i$, with $i = 1, ..., N$. On the liability side, interbank borrowings are denoted by b, with the borrowings of an individual bank being the sum over i rows:

$$b_j = w \sum_{i}^{N} x_{ij}$$

In addition to interbank borrowings, each bank finances itself through deposits, denoted by d. The remainder of the balance sheet, i.e. total assets a minus interbank credit b and deposits d, represents the bank's equity, denoted by c, which can be thought of as Tier 1 capital under the Basel framework. The bank's liabilities as a whole, denoted by l, are thus composed of the bank's equity c, deposits d and interbank borrowings d. Thus, the liability structure of bank i can be denoted as l_i=c_i+d_i+b_i, with i=1,...,N. A bank is solvent where the difference between assets, i.e. loans made out to banks and the real economy, and debt financing received, both from other banks and depositors, is positive (see [7]):

$$c_i \equiv (e_i + i_i) + (b_i + d_i) \geq 0$$

To set the bank's external assets, a two step process is applied: (i) an overhang in interbank borrowing is levelled out by first setting e_i=b_i-i_i and (ii) by evenly spreading out the remainder of aggregate external assets E left after step (i) over all banks. Step (ii) is executed by taking the difference between E and the sum of external assets already distributed in step (i), denoted as $\tilde{e}$, and dividing between the number of banks N. Thus, the amount of external assets distributed in step (ii), denoted as $\hat{e}$, can be formally expressed as:

$$\hat{e}_i = \frac{\left[(E - \sum_{i=1}^{N} \tilde{e})\right]}{N}$$

With the determination of external assets, the asset side of the balance sheets is complete. The only missing elements are therefore found on the liability side, namely bank capital c and deposits d. Bank capital c is set as a fixed proportion γ of the bank's assets, i.e. $c_i\gamma \times a_i$. Deposits are then set as the residual, namely $d_i a_i - c_i - b_i$.

4.2 Network topology

For the actual generation of the interbank lending network, we have looked at two different types of networks: Erdős–Rényi random networks (as proposed by Nier et al. [9]) and scale free networks, whose degree distributions follow a power law.

4.2.1 Erdős–Rényi Random Network

Introduced in 1959 [6], the Erdős–Rényi random network is generated under the assumption that the probability that one bank has lent to another shall be fixed for all pairs (i, j): each directed edge is included in the network with a probability p_E, and this is done independently from any other existing edges. This implies that both the incoming and the outgoing degree distributions of the Erdős–Rényi random interbank lending network follow a binomial distribution.

The function that generates this network, *random_graph.m*, had already been written by the previous group that worked on this network [1]. In essence, it creates an empty $N \times N$ network and draws N^2 uniform random numbers between 0 and 1 and then rounds up or down to the first integer in order to decide whether or not an edge is present. After this is done, the diagonal is set to zero as loops do not exist in our models.

4.2.2 Scale Free Network

While the Erdős–Rényi random graph interbank lending network is a good first model, empirical data suggests that the degree distributions in interbank lending networks are power-law distributed, i.e. a few (mainly large) banks lend to many banks and many small banks lend to only a few other banks. In particular, Boss et al. [4] have empirically analyzed the network topology of the Austrian interbank market and have found that the degree distribution follows a power-law distribution. In addition, Markose [8] has empirically found power law distributed derivatives exposures in the OTC derivatives market, a subsegment of the global interbank markets. While Nier et al. [9] have already noted that real-world banking networks do not conform with the Erdős–Rényi random graph networks and have therefore implemented a two tiered banking network with small and large banks, they have not tested their contagion alogrithm on a scale free interbank network. We go one step further by implementing a Barabasi and Albert [2] style preferential attachment algorithm to generate a power-law distributed network.

In order to construct this type of network, we chose to implement an algorithm for directed preferential attachment as described by Chung and Lu [5]. It is an algorithm with three main steps: the I step, the O step and the R step, each of which occurs with their own probability p_I, p_O and p_R:

I: A new node v is added to the network, and a directed edge is added towards a previously existing node u. Node u is picked with a probability dependent on its incoming degree.

O: A new node v is added to the network, and a directed edge is added from a previously existing node u. Node u is picked with a probability dependent on its outgoing degree.

R: A directed edge is added from existing node s to existing node t, where s and t are chosen with a probability dependent on their outgoing and incoming degree respectively.

We implemented this in MATLAB with the function *pref_graph.m*, a function with three inputs: the number of banks N, the probability of step O, p_O, and the probability of step I, p_I. We set $p_R = 1 - (p_I + p_O)$ so that the user chooses this implicitly. Upon initialization, we begin with a 1×1 empty matrix. Then, one of the three steps is chosen based on the input probabilities using the MATLAB function *randsample*. In the case of step I, a new empty row is added with length equal to the existing amount of columns. This row represents the new bank v. In order to determine existing bank u, we pick a column using *randsample*. This time it is weighted by the sums of each column, which is a measure of the number of incoming edges. We then set matrix element (v, u) to 1. Step O follows the opposite procedure: we add a new empty column with length equal to the existing number of rows, which represents the new bank v. We then choose the existing bank u weighted by the sums of each row, and we set (u, v) to 1. Finally, in step R we choose a row s weighted by the sums of the rows, and a column t weighted by the sums of the columns. We then set (s, t) to 1.

Up to this point the amount of banks N has not been used. We use this as our stopping condition: once either the amount of columns or rows is equal to N, the above procedure halts. The matrix is then appended with empty rows or columns (the one which is not yet equal to N) until the matrix is square. An alternative would have been to simply remove either the O or the I step from the process until the matrix is square, but no preference was specified in the literature. We then proceed to take out all diagonal elements, as our models contain no loops.

In order to establish that we indeed implemented an algorithm that produces a scale free network, we numerically analysed both the incoming and outgoing degrees of our preferential attachment network. This was done by counting the number of incoming and outgoing connections, and analysing their frequencies. Figure 1 shows both the normal scale and the log-log scale degree distributions for $N = 50$ banks, taking $p_O = p_I = 0.08$. We have also produced these distributions for networks of larger sizes such as $N = 200$. At these sizes the power law distribution becomes smoother with fewer deviations, but for such N the computation times of the performed simulations were simply too high.

4.2.3 Comparing the networks

We have now established how we construct both the random network and the power law network. However, these networks are built on very different parameters: the first uses the

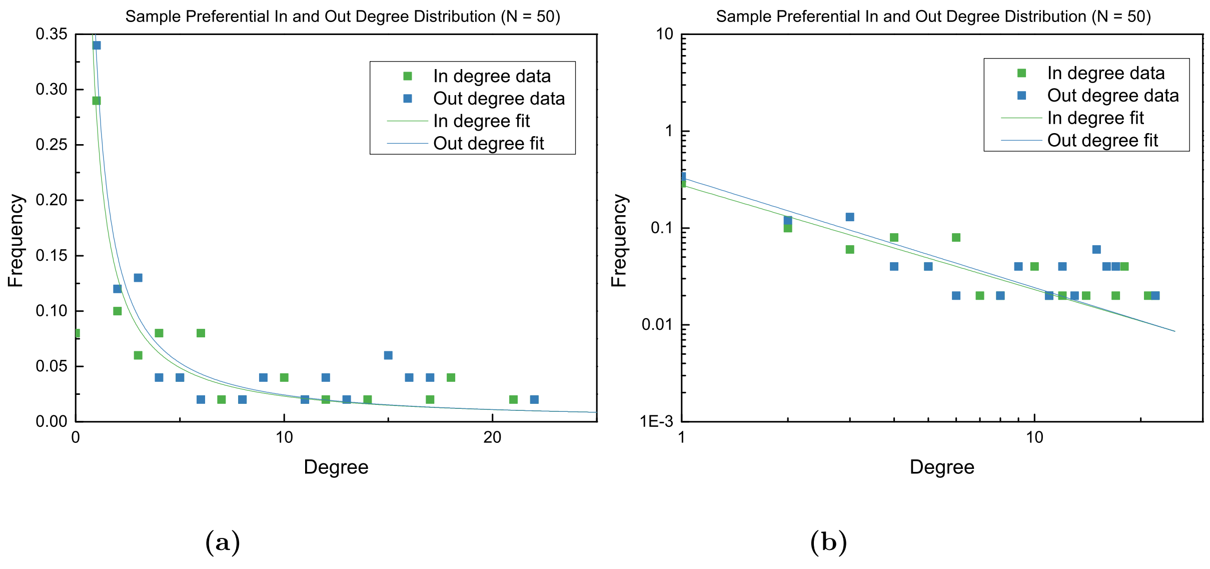


Figure 1: *These two figures show the degree distribution of the network generated using the preferential attachment algorithm described in section 4.2. Generated for $N = 50$ and $p_I = p_O = 0.08$, the degree distribution were fitted with a power law distribution $y = ax^b$. The in and out degree fits converged with an adjusted R^2 of 0.917 and 0.911 respectively, with $b = -1.08$ and $b = -1.13$.*

Erdős–Rényi probability p_E, while the second uses two parameters, one for a new incoming connection p_I and one for a new outgoing connection p_O. At first sight there is no clear link between these parameters, as they fulfil a very different task. Nevertheless it is true that in both models choosing the same parameter values should result in a network with similar global properties. One of these is the expected number of links in the network, $\langle m \rangle$. In the case of the random network, this is readily computed: we have a network of size $N \times N$, where each element has a probability p_E of being 1, a link. Therefore we can say that $\langle m \rangle = N^2 p_E$, or equivalently that $p_E = \frac{\langle m \rangle}{N^2}$.

For the preferential attachment network this is not as trivial. An analytical expression for $\langle m \rangle$ was not found in the literature used to construct our specific network, so instead we opted for a numerical approach. In order to do so, we first made one critical adjustment to the model: from this point on we use that $p_I = p_O$ so that our preferential network has only 1 parameter we can vary, which we will refer to as p_P. This simplifies the problem of connecting the two networks to finding the relationship between p_E and p_P, instead of between p_E and both p_I and p_O. This results in identically distributed incoming and outgoing degrees, which is not necessarily true empirically.

Having made this adjustment, we now used a numerical approach to estimate $\langle m \rangle$ in the preferential attachment network. As for our other simulations we use $N = 50$ banks, and we simulate a preferential network for various values of p_P 100 times. We then calculate both the mean value and the standard deviation of the number of links m. Having previously established that $\langle m \rangle = N^2 p_E$, these quantities give us a numerical relationship between the parameters of both models. As shown in figure 2, we found that this relationship is well described by a double exponential decay of the form $y = y_0 + A_1 e^{-x/t_1} + A_2 e^{-x/t_2}$. Although we cannot explain the nature of this double exponential relationship, it gives us the ability to choose a parameter value for both networks such that we expect them to have the same number of links, giving us a measure to compare both networks.

Ideally we would like to choose $p_E = 0.2$ like used in last year's project, in order to compare our results to theirs. However, this would mean $p_P \approx 0.025$, in which case we no longer produce a scale free degree distribution. This is because the network will saturate due to the R step of the algorithm. Instead we take $p_P = 0.08$, which corresponds to $p_E = 0.1$. As shown in figure

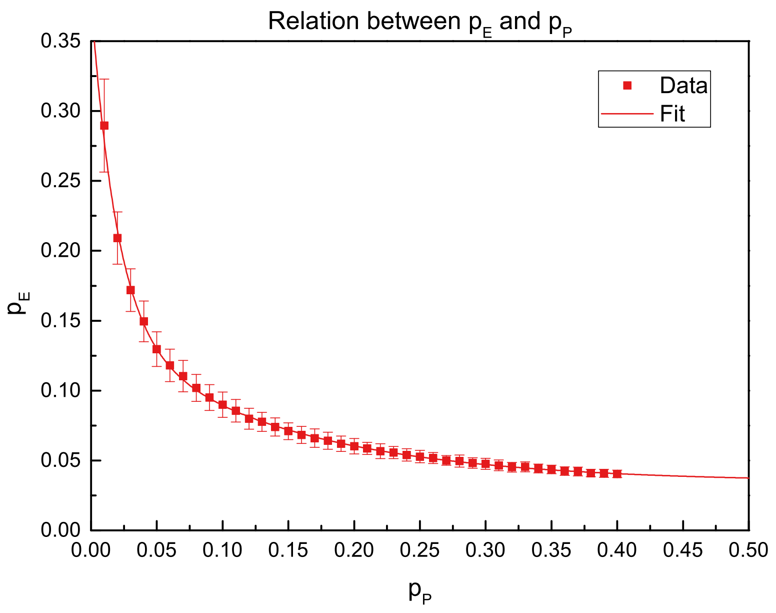


Figure 2: *A double exponential fit of the numerically computed relationship between p_P and p_E, based on $p_E = \frac{\langle m \rangle}{n^2}$. These results were obtained from creating 100 networks for every value of p_P and averaging over the results to find $\langle m \rangle$. The error bars shown are the standard deviations. The fit shown here converged with an adjusted R^2 value of 0.998.*

1, this still leads to degree distributions that are well fit with a power law distribution.

4.3 Randomizing the Assets

In the balance sheet described in section 4.1, a bank's interbank borrowings b_j were simply the product of the number of outgoing connections and the loan weight w. In reality however, not every link would have the same value, as not every loan will be for the same amount. In order to take this into account, we do not only compare the random network to the scale free network under these well-behaved conditions, but also under more variable circumstances. To do so, we first calculate b_j as we would in the standard case, but we then randomize each value to obtain a new value for the interbank borrowings, $\bar{b}_j = r_j b_j$ with $r_j \in [r_{min}, r_{max}]$ picked uniformly. However, we do not want the total amount of assets in the system A to change, and thus we also enforce that the r_j are chosen such that $\sum_j r_j b_j = \sum_j (b_j) = B$.

This process is captured by the function *generate_banks_randomized.m*, and it uses a procedure designed by us. We first calculate the b_j like we would have in the original generation function and calculate the total amount of interbank borrowing $\sum_j (b_j)$. After this we draw N values of $r_j \in [r_{min}, r_{max}]$ and compute all $\bar{b}_j$.

We then follow an algorithmic approach in order to satisfy $\sum_j \bar{b}_j = B$. The algorithm has one main stopping condition, and that is if $\sum_j \bar{b}_j$ falls within the range $[(1-\epsilon)B, (1+\epsilon)B]$, where ϵ is an input threshold parameter. While this is not the case, there are two scenario's, the first of which is $\sum_j \bar{b}_j < (1-\epsilon)B$. In this scenario we run a for loop that consecutively redraws every $r_j < 1$, continuously checking the stopping condition. This continues until the sum falls in the desired range, which happens in finite time as long as $r_{max} > 1$. The other scenario is $\sum_j \bar{b}_j > (1+\epsilon)B$. In this case we do the opposite, redrawing values $r_j > 1$ until the stopping condition is met, which also happens in finite time as long as $r_{min} < 1$. The result is what we wanted: a set of randomized values $\bar{b}_j$, for which $\sum_j \bar{b}_j \in [(1-\epsilon)B, (1+\epsilon)B]$.

One way to interpret this randomization is as randomizing the weight w for each individual bank, creating a vector w_j. This can be seen from the earlier definition of b_j, where we said that it was found by multiplying w with the sum of each column. If we now divide $\bar{b}_j$ by the sums of each row, we find a new value w_j for each bank. This means that in essence we have changed the system such that each bank lends out it's own characteristic amount of assets, instead of this being the same for every single bank. It is dependent on the number of incoming edges, but in a randomized fashion. In reality the value of w_j would also depend on which bank one is lending to. One would have to construct a matrix $w_{i,j}$, but this is not captured in our model at this time.

Now that we have a new set of values $\bar{b}_j$, we compute a set of interbank assets $\bar{i}_j$ by multiplying the sums of the rows with the respective value of w_j, the vector. After this is done, we compute the other relevant quantities (e, a, c, d) as described in section 4.1.

4.4 Regulation

Due to the randomization of the banks assets described in section 4.3, some banks will be more resilient to shocks than they would have been using the balance sheet from section 4.1. Conversely, others might be worse off, potentially posing a threat to the stability of the network. It is for this reason that we introduce the concept of a regulator, that evaluates a banks total interbank borrowings as a function of the number of incoming edges. As a benchmark value it uses the standard rule for interbank borrowing b_j, which is normally given by multiplying w (the scalar, not the randomized vector) by the sum of the number of incoming edges. The regulator then compares b_j to the randomized $\bar{b}_j$ by checking if $\frac{\bar{b}_j}{b_j} > \alpha$, where α is the regulation parameter. If this is indeed the case, it decides if certain borrowings should be disapproved for the sake of the stability of the network. A different interpretation would be that a bank has to get approval from the regulatory agent before this specific bank can borrow from another bank.

The exact nature of deciding which and how many borrowings should be disapproved is open to interpretation, and we decided to test two distinct approaches. The first one we call the random regulator, and it is contained in the function *RandRegulator.m*. It chooses which bank to test first by iterating through a random permutation of all banks. If for this bank it is not true that $\frac{\bar{b}_j}{b_j} > \alpha$, it continues to the next randomly chosen bank until it finds one for which it is. For this bank the regulator goes through all borrowings and randomly chooses one to terminate. This removes the incoming connection from the network, causing the banks interbank borrowing to go down by the value of that loan w_j, and thus by extension also its external assets. This in turn causes the external assets of the bank who's loan was terminated to rise by the same amount, while their interbank assets decrease. As the number of borrowings in the network has decreased, the regulator also recalculates its benchmark value b_j, before starting the above procedure again. The procedure is repeated until either $\frac{\bar{b}_j}{b_j} \leq \alpha$ for all j, or if the iteration number exceeds 1000. In the few situations we tested no more than 50 iterations were required before the regulator had finished making changes, but as the function is not computationally expensive we choose to be on the safe side.

The second regulator we introduce attempts to do a better job than the random regulator by going through the network in an organized fashion. We call this regulator the smart regulator, contained in the function *SmartRegulator.m*. Although it uses the same benchmark condition as the random regulator, it is different in several ways. To begin with, all banks are not treated equally: the regulator evaluates $\frac{\bar{b}_j}{b_j}$ for all j and then sorts the banks from highest to lowest ratio. It does this so that it first attempts to regulate the bank whose interbank borrowings exceed the benchmark value by the most, as a default in one of these banks leads to large losses in other banks. In order to avoid helping banks that have no borrowings, their ratio is set to 0, as for these banks $\frac{\bar{b}_j}{b_j} = \frac{0}{0}$. This is equal to NaN in MATLAB, and NaN is considered larger than all real numbers.

The way in which the regulator decides what borrowings to terminate is also different from the random regulator. The smart regulator now sorts the banks by their ratio $\frac{\bar{e}_j}{e_j}$, which is the ratio of their external assets after the randomization of b to the value of e that would be obtained in the default bank generation. The sorting is done in an ascending manner, so that the banks that have the least amount of external assets compared to what the regulator would expect them to have will get their loans terminated first. Once the regulator has computed these ratios, he begins iterating through the network: starting with the bank with the highest $\frac{\bar{b}_j}{b_j} > \alpha$, he goes through their borrowings with the other banks ordered by ascending $\frac{\bar{e}_j}{e_j}$. Once the first incoming connection is found, it is again removed from the network, and the values of interbank borrowing, lending and external assets are again updated, as well as the benchmark values. The same procedure is started again, beginning from the calculation of $\frac{\bar{b}_j}{b_j}$. Like with the random regulator, the procedure is repeated until either $\frac{\bar{b}_j}{b_j} \leq \alpha$ for all j, or if the iteration number exceeds 1000.

4.5 Shock generation and propagation

Once the banking ecosystem has been generated, given a (possibly randomized) balance sheet and potentially regulated, we test the stability of our network. To do so, we generate a shock, denoted as s_i by initially wiping out a fraction of a single bank's external assets. Following the seniority rules of the capital structure and in line with an orderly bankruptcy/liquidation procedure, the loss generated by the shock to the external assets will first be absorbed by the bank capital c_i, in a second instance by interbank creditors b_i and in the last instance by (partially deposit insured) depositors d_i. If the shock wipes out the equity, i.e. $s_i > c_i$, then the interbank loss IB will be borne equally by that bank's interbank creditors, with k_i denoting the number of creditor banks of bank i. Formally, where $(s_i - c_i) < b_i$ the share of loss absorbed by the interbank creditors will be:

$$IB = \frac{(s_i - c_i)}{k_i}$$

The residual of the loss after absorption by the interbank creditors will be borne by depositors. In this model setup, shocks are propagated through the system via the interbank credit markets only, both shareholders and depositors effectively absorb exogenous shocks without contributing to shock contagion themselves. Where a bank receives a loss through a default of the outstanding interbank loan to the defaulting bank, the loss will be cascaded through the capital structure and, if large enough, in turn be transmitted to the banking ecosystem via interbank loans.

The function that governs the shock process is called *simulate.m*, and it was written by the previous group that worked on this project [1]. We have however made a number of small corrections to the code, and some slight adjustments for the case of the randomized balance sheet, contained in the function *simulate_randomized*. The function is rather complex and consists of multiple stages, and as no description was available in the previous report or code we have put an elaborate description in the Appendix, section 7.

5 Results and Discussion

We have performed three distinct types of simulations: comparing the random and scale free network for the standard balance sheet of section 4.1; comparing the random and scale free network for the randomized balance sheet of section 4.3; and comparing the random and scale free network for the random and smart regulator from section 4.4. Most of these simulations were done using either the set parameter value or the same range of parameter values, and these are listed in table 1. In the cases where we use different parameters, this is explicitly mentioned in the figure captions. In all of the figures the thick line represents the mean value of the number of defaults, while the shaded region depicts the standard deviation.

Table 1: Main parameter values

Parameter	Description	Benchmark value	Range of variation
E	Total external assets	100 000	Fixed
S	Total Shock Size	100 000	Fixed
$Runs$	Number of repetitions for each simulation	100	Fixed
N	Number of banks in the network	50	Fixed
p_E	Erdos−Renyi probability	0.1	Fixed
p_P	Preferential Attachment parameter	0.08	Fixed
r_{min}	Minimum randomization factor	0.5	Fixed
r_{max}	Maximum randomization factor	2	Fixed
α	Regulation parameter	1.2	1-2
Θ	Percentage of interbank assets to total assets	20%	0-50%
γ	Percentage of net worth to total assets	1%	0-10%

5.1 Standard balance sheet

We first simulate both the random and the scale free network for various values of Θ and γ, using the parameters in table 1. Keeping Θ fixed and varying γ, we obtain the results displayed in figure 3. In this figure we see that the random network exhibits a rapid decline in number of defaults, before transitioning into a much slower decline. On the other hand, the scale free network exhibits a smooth, exponential like decline, and outside of a small region of γ values it has less defaults than the random network. The variance of both networks is small, showing very regular behaviour with almost no fluctuations in the number of defaults.

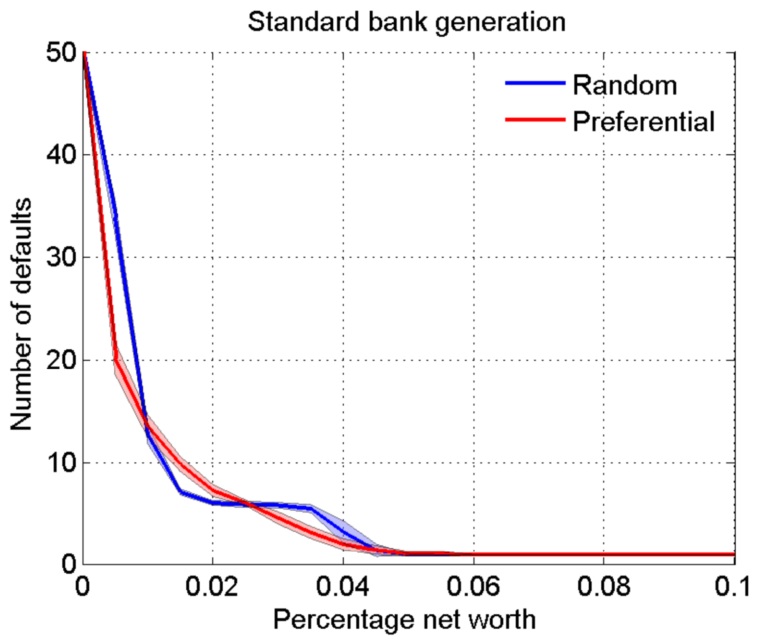


Figure 3: *The figure presents the relationship between the number of defaults and the banks' percentage net worth for both the random and the scale free network. Here we made use of the standard balance sheet.*

Next we vary θ (the percentage of interbank assets in total assets) for three different values of the banks' net worth: $\gamma = 0.01, 0.03, 0.05$. This leads to figure 4. For $\gamma = 0.03$ and 0.05 we get slightly better behaviour in the scale free network, starting from $\theta = 0.1$. For $\gamma = 0.01$ however, we observe a substantial improvement. For $\theta = 0.5$, we have about half as many defaults in the scale free network than we have in the random network.

We can hypothesize why this is true. To do so, we think of the degree distributions of the two networks. In the case of the power law network, there is a very large number of banks that have only a few connections, which means that in the case that they are shocked, most of the network will not be affected by this. Especially in the case of low values of γ, where banks are very vulnerable to being shocked, this lower average level of connectivity might result in a safer network. For higher values of γ moderately connected banks might still have enough capital to survive the propagating shock, offsetting the effect of having many potential links to banks that get shocked. Moreover, the highly connected banks can act as a buffer for shocks, reducing the amount spread through the system. Together, this could explain why the scale free network appears to be more resilient than the random network in the case of standard balance sheet generation.

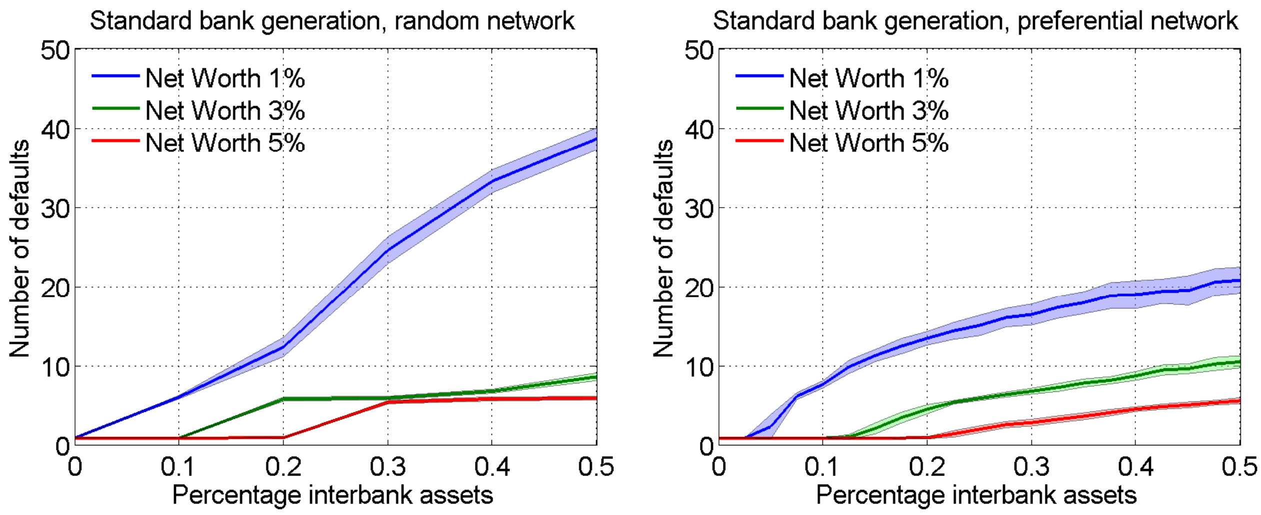


Figure 4: *The figures present the relationships between the number of defaults and the percentage of interbank assets in total assets in the network for various values of γ. Figure a depicts the random network, while figure b depicts the scale free network. Here we made use of the standard balance sheet.*

5.2 Randomized Assets

Here we again first simulate both the random and the scale free network for various values of Θ and γ, using the parameters in table 1. This time however the balance sheet is not the standard variant, but the one randomized as described in section 4.3. Keeping Θ fixed and varying γ, we obtain the results displayed in figure 5. We immediately notice a few differences compared to figure 3. Both networks exhibit a far larger variance, and while the random network has mostly retained its previous declination pattern, the power law network has changed a lot. It has lost its apparent exponential decline, showing much more irregular behaviour is no longer monotonically decreasing. Moreover, the scale free network is now worse than the random network for values of γ up to 0.4.

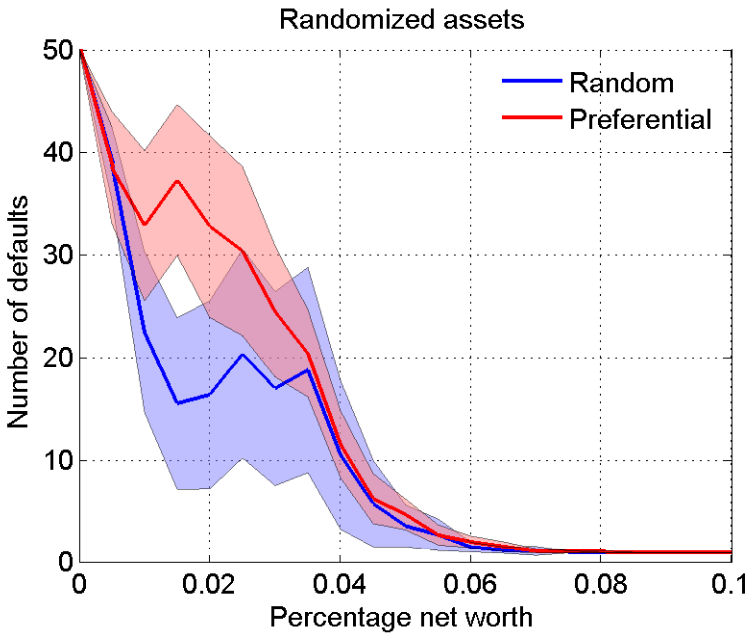


Figure 5: *The figure presents the relationship between the number of defaults and the banks' percentage net worth for both the random and the scale free network. Here we made use of randomized balance sheet.*

Next we again vary Θ (the percentage of interbank assets in total assets) for three different values of the banks' net worth: $\gamma = 0.01, 0.03, 0.05$, but now with a randomized balance sheet.

This leads to figure 6. Here we see similar features as when we looked at γ: both the scale free and the random network are worse than they were in figure 4, but it is again the case that the scale free network is now worse than the random network, whereas it was the opposite before. This is especially clear for $\gamma = 0.03, 0.05$ and $\Theta = 0.5$, where on average the scale free network has 10 more defaults than the random network. It should be noted however that the random network exhibits a larger variance than the power law network.

We can again hypothesize about why we observe these features. The higher variance in both the scale free network and the random network can be attributed to the fact that there is now an inherent randomness in the balance sheet: over many runs, various different interbank borrowings will be generated, leading to a wide variety of situations. Why the random network would do better in such an environment than the scale free network could be thought of in the following way. Because of the way the networks are structured, each bank in the random network is randomized in approximately the same way, as they all have similar degree distributions. This means that on average the system remains similar to what it was before the randomization, with the occasional negative effect that some banks borrow far more than they should.

However, in the case of the power law the randomization is not evenly distributed among the network: the small banks that had little capital to begin with will still have relatively little, remaining vulnerable to shocks. But the real difference lies with the highly connected banks: for these banks the change in interbank borrowings can be much larger on the absolute scale, leading to more extreme behaviour. In the event that such a highly connected bank becomes relatively unhealthy, this could have large consequences for the stability of the network. These events are more common in the scale free network than in the random network, which could be an explanation for the behaviour we observe.

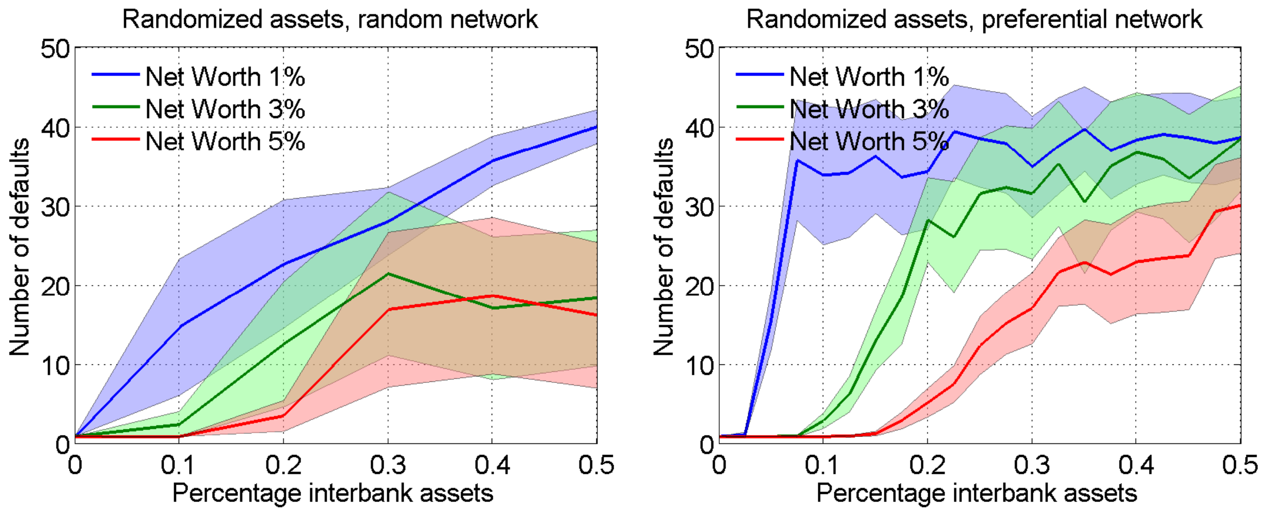


Figure 6: *The figures present the relationships between the number of defaults and the percentage of interbank assets in total assets in the network for various values of γ. Figure a depicts the random network, while figure b depicts the scale free network. Here we made use of the randomized balance sheet.*

5.3 Regulation

In this section we look at the behaviour of the two networks under regulation from both the random and the smart regulator, after the networks have undergone interbank borrowing randomization. In the first scenario we compare the random regulator to the smart regulator, as can be seen in figure 7. Here we used the standard parameter values, except that we chose $\gamma = 0.005$ as this resulted in approximately the same number of defaults for both networks in the randomized, unregulated case. We observe that the smart regulator does better than the random regulator for almost all the values of α, for both the random and the preferential network.

We can then ask the question of whether or not our regulators are improving the stability of the networks. For large values of α (such as 2), virtually no regulation will be done: $r_{max} = 2$, which makes it very unlikely that any of the $\underline{b}_j$ is so much larger than the benchmark b_j. At these values of α we see that the number of defaults is higher than for low values of α in the case of the smart regulator, while in the random regulator they remain virtually equal. It thus seems as if the random regulator is hardly improving the stability of the network, even though it is reducing interconnectivity, making it harder for shocks to spread. The smart regulator on the other hand is bringing the number of defaults down, albeit in moderate amounts. We hypothesize that this is due to a combination of decreased interconnectivity, as well as because of the fact that it prioritizes regulating those borrowings that are the least beneficial to the network.

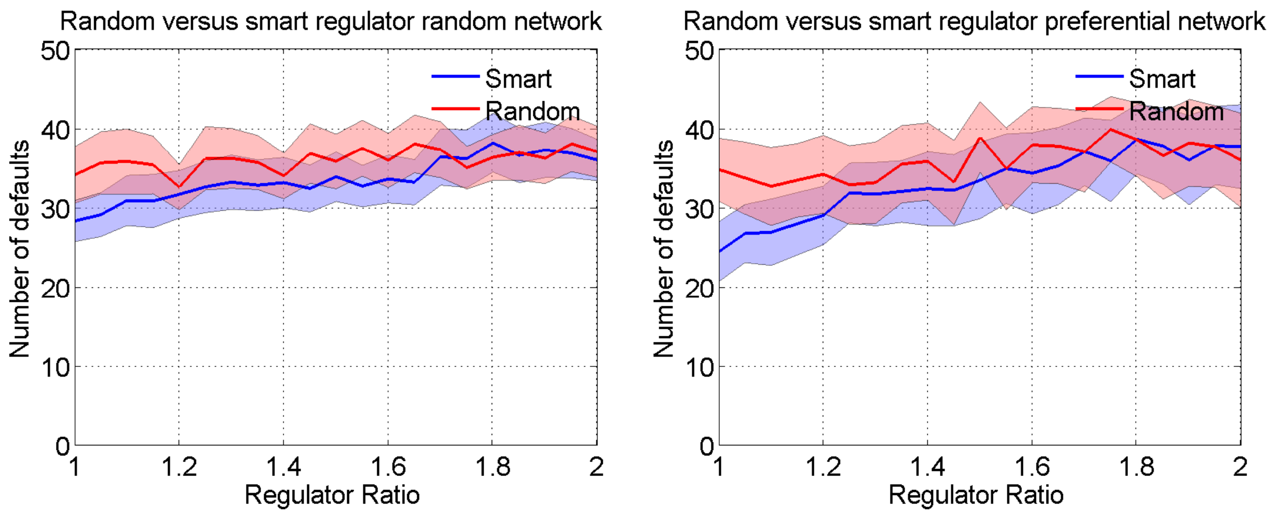
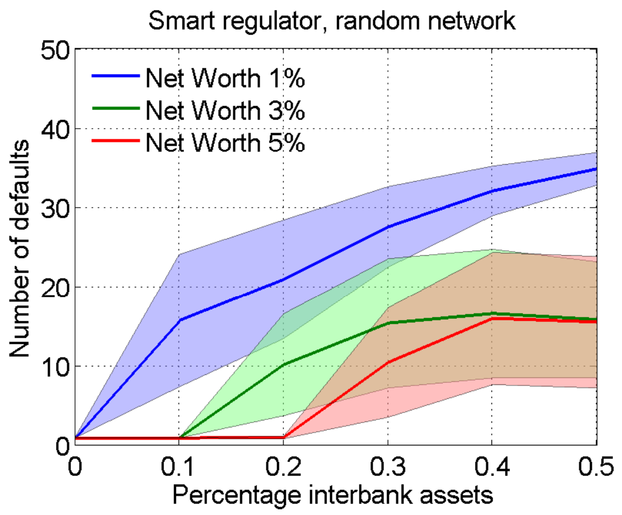


Figure 7: *In these figures we compare the random regulator to the smart regulator for both the random and the scale free network. Here we used the standard parameter values, except that we chose* $\gamma = 0.005$ *so that both networks started off at approximately the same number of defaults.*

As the smart regulator has shown itself to be superior to the random regulator, we now look at how it behaves under various values of Θ and γ as depicted in figure 8. In the random network we get a light improvement compared to the randomized unregulated version of figure 6. The variances of our results are smaller, which would be because the regulator is reducing the disproportionally large interbank borrowings. All in all we observe that the network is indeed more resilient for all the values of net worth γ we consider. In the preferential network we get a much larger improvement for all the values of net worth and percentage of interbank assets we consider. We can try to explain this in much the same way as how we explained why the scale free network behaved so much worse under randomization: banks with large amounts of interbank borrowings undergo the biggest absolute changes in b_j under the randomization. If these are indeed the reason that the scale free networks behaved so poorly when randomized, then regulating these specific banks will in turn result in a substantial improvement of the networks stability. The regulator would thus have the potential to stabilize the network more in the case of the scale free network, as these large banks are uncommon in the random network.



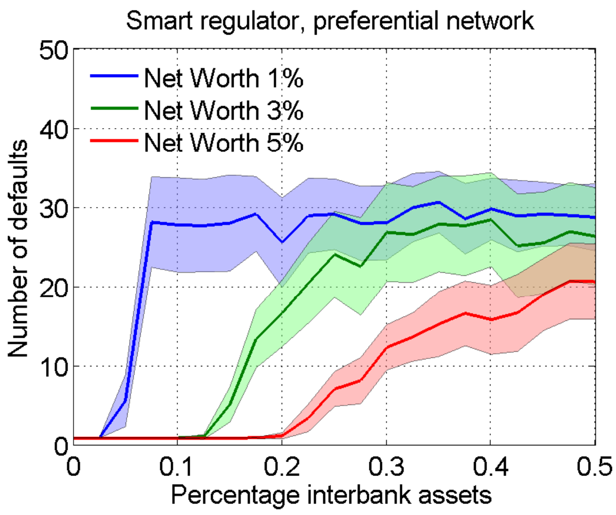


Figure 8: *The figures present the relationships between the number of defaults and the percentage of interbank assets in total assets in the network for various values of γ. Figure a depicts the random network, while figure b depicts the scale free network. Here we made use of the randomized balance sheet and the smart regulator.*

6 Summary and Outlook

We have built on the existing literature on shock contagion in interbank networks, namely Nier et al. [9] and have extended this by (i) making the random network used in the shock contagion process conform more closely to a real world setting with power-law distributed degrees, (ii) randomizing the interbank borrowing during the balance sheet generation and (iii) by introducing a regulatory agent that analyzes and regulates a measure of interconnectedness of banks.

In summary, the results show that using a scale free interbank loan network results in an overall more resilient banking ecosystem when using the standard balance sheet generation procedure. This finding could be explained in a number of ways. This could be because having a two tiered banking ecosystem with a few highly connected banks and many sparsely connected banks can go either way. On the one hand, if the highly connected bank is hit, the shock is propagated to many interbank lenders, thereby potentially affecting a larger portion of the banking system. On the other hand, since the shock can be distributed among many banks, the individual shocks are small and less likely to lead to a default. In this sense the many creditor banks act as stabilizers to system. Moreover, if one of the many less connected banks will be hit, the shock that is propagated will only be spread to a few other banks, but it is likely to be larger, since it is spread over fewer creditor banks. This is in contrast to the random banking network, where most banks have a moderate degree of connectivity paired with an average ability to withstand shocks.

The introduction of the randomization lead to a larger degradation of the scale free networks than it did in the random networks. We hypothesize that this is due to the fact that the most interconnect banks undergo the largest absolute changes under randomization, and since these are more explicitly present in the scale free networks, these effects are most present here compared to the random network.

Furthermore, we found that the introduction of a regulator that attempts to alleviate the negative effects of the randomization can lead to an improvement of network resilience. This was only true in the case of the regulator that made an educated decision about which connections to terminate; our results indicate that a random regulator which simply reduced the interconnectivity of the network did not improve the situation, whereas the smart regulator led to lower numbers of defaults.

The changes made to the network in our project is just another step in a series of possible alterations that can be made in order to extend and refine the analysis of contagion risk in interconnected banking systems. Further changes and questions that could be analyzed in the future are:

- Introduction of a lender of last resort, i.e. the government or the central bank, which enters and halts the default cascade where the second layer in the failure cascade indicates that the effects of the shock will be too large.
- Adding a further refinement to the default mechanism to make it conform more to a real world setting. For example, bank defaults will always lead to concurrent bank runs by the depositors and fire sale of assets, which will in turn lead to further losses of the defaulting banks. This effect is not yet captured in the model, but could be integrated as a shock multiplier.
- The balance sheet structure could be further extended so as to include different types of assets. Shocks could then be inflicted concurrently on different banks and different asset classes.
- Lastly, the regulator could be made even smarter, by having it focus concurrently on both the interconnectedness and the Tier 1 capital of the bank. The relationship between regulating by inter-connectivity vs. economic capital could be analyzed.

References

[1] Systemic risk in network model of financial shock spreading. *Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB*, 2014.

[2] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] IMF BIS, FSB. Guidance to Assess the Systemic Importance of Financial Institutions, Markets and Instruments: Initial Considerations. *Report to G-20 Finance Ministers and Central Bank Governors*, August 2009.

[4] Michael Boss, Helmut Elsinger, Martin Summer, and Stefan Thurner. Network topology of the interbank market. *Quantitative Finance*, 4(6):677–684, 2004.

[5] Fan Chung and Linyuan Lu. Complex graphs and networks (cbms regional conference series in mathematics). 2006.

[6] P. Erdos and Renyi A. On Random Graphs. *Publicationes Mathematicae*, 6:290–298, 1959.

[7] Andrew G. Haldane and Robert M. May. Systemic risk in banking ecosystems. *Nature*, 469(7330):351–355, January 2011.

[8] Sheri Marina Markose. Systemic risk from global financial derivatives; a network analysis of contagion and its mitigation with super-spreader tax. *IMF Working Paper*, (12/282), 2012.

[9] Erlend Nier, Jing Yang, Tanju Yorulmazer, and Amadeo Alentorn. Network models and financial stability. *Journal of Economic Dynamics and Control*, 31(6):2033 – 2060, 2007.

[10] Christian Upper. Using counterfactual simulations to assess the danger of contagion in interbank markets. *BIS Working Papers*, (234), August 2007.

7 Appendix

7.1 Description of basic shock generation and propagation code

Shocks are propagated through the banking ecosystem by the single_shock function, which is a nested function within the simulate function. The basic setup is provided through a copy of the $N \times N$ adjacency matrix (B in the script), which is multiplied by the loan weights. This is referred to as interbank_left in the script, since the defaulting loans will be removed in the matrix and only non-defaulting loans will remain:

```
 interbank_left=B*w;
```

Further to that, a vector of all banks is created where the remaining shock to be applied to the respective bank at each stage is updated as the shock propagates through the system:

```
remaining_shocks=zeros(1,N);
```

The externally set shock is then inserted into shock vector of the first bank to be shocked via:

```
remaining_shocks(s)=S;;
```

Once the shock vector and the adjacency matrix are set up, the single_shock function is then called and run once for every bank in the banking ecosystem through a for loop which runs for as many times as there are banks:

```
for s=1:length(remaining_shocks)
[i,c,d,remaining_shocks,interbank_left]=single_shock(a,c,s,d,i,
  interbank_left,remaining_shocks,w);
end
```

The single_shock function begins with wiping out bank capital, as described in the model. It does so by calling a further function, absorber:

```
[c(s),postnet_shock]=absorber(c(s),remaining_shocks(s));
```

The absorber function takes a certain level of capital and a certain shock level as input and returns the either the remaining capital (if its not fully wiped out) or the residual loss (referred to as postnet_shock in the code) if loss is larger than the capital:

```
function [capital1,rest] = absorber(capital0,loss)
    capital1=max(0,capital0-loss);
    rest=-min(0,capital0-loss);
end
```

Once the equity has thus been wiped out in the single_shock function, the interbank credit will be wiped out either fully (where $s_i \geq b_i$) or partially (where $s_i < b_i$). The interbank borrowing entry of the shocked bank is thus reduced by either the smaller of either (i) the shock amount (if $s_i < b_i$) or (ii) by the full interbank credit amount:

```
interbank_shock=min(postnet_shock,b(s));
b(s)=b(s)-interbank_shock;
```

The shock is then spread to the other banks by calculating the interbank loss IB (see 4.5). For this, a first step is to establish the number of outstanding creditor banks the shocked bank has. This is achieved by summing up the respective column of the adjacency matrix that represents the interbank debt of the respective bank and dividing it by the uniform link weights w:

```
num_creditors=sum(interbank_left(:,s))/w;
```

Once the number of creditors has been established, the maximum interbank loss IB is calculated as the residucal loss (postnet_shock) divided by the number of creditors (num_creditors). Concurrently, a shock variable individua_interbank_shock is introduced which establishes whether the interbank debt of the creditor banks should be wiped out in full or only partially in the amount of the interbank loss IB:

```
individual_interbank_shock=min(interbank_left(:,s),postnet_shock/
  num_creditors);
```

The interbank loss IB is then applied to the interbank loan column in the matrix and is spread through the interbank system by hitting the vector that sets out the capital for every bank c:

```
interbank_left(:,s)=interbank_left(:,s)-individual_interbank_shock;
[c,new_shocks]=absorber(c,individual_interbank_shock);
```

What the residual loss that will then hit interbank creditors in the third layer, i.e. creditor banks of creditor banks of the originally shocked bank, is then recorded in the remaining shock vector and will then be applied in the next runs of the for loop.

```
remaining_shocks=remaining_shocks+new_shocks;
```