University of Oslo

FYS3150

Computational Physics

Project 5 - Monte Carlo Modeling of Transactions

Author:
Vebjørn Gilberg

Author:
Trond-Wiggo Johansen

April 24, 2017

Contents

Abstract

The statistical character of the wealth of individuals in a stable economy was investigated by the italian social economist Vilfredo Pareto more than a century ago using an inverse power law that we implement and analyze. The distribution of money in a closed system where the financial agents involved are making transactions with one another is analyzed by using the Gibbs distribution, but when more criteria is included to make the transactions more realistic we find that the distribution no longer follows that given by Gibbs, but rather a γ-distribution. The impact of micro-economics on the macroscopic economical system is also studied as well as the tail of the probability density function obtained from the simulation which follows the power-law given by Vilfredo Pareto.

Github repository:
`https://github.com/wiggoen/FYS3150/tree/master/Project_5`

1 Introduction

The stock market can be unpredictable, and therefore a probabilistic method is needed in order to simulate it. We simulate a situation in which a series of transactions take place between people called financial agents by using Monte Carlo methods. The higher end of the distribution of money follows Pareto's law $\omega_m \propto m^{-1-\alpha}$ [1] where the constant α takes a value $\alpha \in [1, 2]$ and m is the income of the agents. These transactions should reflect human relations, and for that reason there's a need to include parameters that can reflect these aspects as well. We start with a simple transaction system, expanding it to be more realistic by introducing new criteria. There is a higher probability that the agents will trade with each other if they have the same income or have traded before. There is a relation between this closed system consisting of micro-economical transactions and the macro-economical distribution of the whole system. We model this distribution at the equilibrium state by using the Gibbs distribution due to the fact that money is conserved in the system. The element of saving some of the money when the transactions takes place is also included and this results in a different distribution than that given by Gibbs.

2 Theory

2.1 Pareto's law and agent-based modeling with random numbers

Given a pair of agents (i, j) with a starting capital of $m_0 > 0$, that can exchange the money at hand, one can extract a distribution that follows Pareto's law [2]

$$w_m \propto m^{-1-\alpha}$$

at the higher end of the distribution of money. The income (money) is represented by m and the Pareto exponent α has a value in the interval $[1, 2]$. In the simple model [3] there are N agents that exchange money in randomly chosen pairs at each step. An important aspect of each transaction is that the money is conserved, that is, agent j's money becomes $m_j^{'}$ and $m_i \rightarrow m_i^{'}$ and the total money before the transaction must the same as the amount after the transaction has taken place

$$m_i + m_j = m_i^{'} + m_j^{'}.$$

To make the transactions random, we introduce a random number $\epsilon \in [0, 1]$ that is extracted from a uniform distribution and multiply it with the initial amount $m_i + m_j$ which yields

$$\begin{aligned} m_i^{'} &= \epsilon(m_i + m_j) \\ \Rightarrow m_j^{'} &= (1 - \epsilon)(m_i + m_j). \end{aligned}$$

An important aspect of this model is that no agents are left in debt, which means that $m \geq 0$. The system eventually reclines toward an equilibrium state which is given by the Gibbs distribution

$$w_m = \beta e^{-\beta m} \tag{1}$$

where

$$\beta = \frac{1}{\langle m \rangle} \qquad \text{and} \qquad \langle m \rangle = \frac{1}{N} \sum_i m_i = m_0$$

with $\langle m \rangle$ representing the average money, m_i the money of each agent and N the number of agents. The result of this is that the agents with $m > m^{'}$, which are the richest ones, decreases exponentially with $m^{'}$. This means that only a few of the agents will remain at the top as far as income is concerned. The vast majority is left with a small amount of money compared to the few agents mentioned above.

2.2 The Gibbs distribution and savings

In a realistic trade the agents would save some of the money they have before making a transaction. In order to simulate this we include a factor λ to represent the fraction saved [3]. This factor will make the Gibbs distribution in equilibrium obsolete. Since the conservation law still holds, the agents i and j now trade an amount of money corresponding to $(1-\lambda)(m_i+m_j)$. With this, we get

$$m_i^{'} = \lambda m_i + \epsilon(1-\lambda)(m_i+m_j) \quad \text{and} \quad m_j^{'} = \lambda m_j + (1-\epsilon)(1-\lambda)(m_i+m_j)$$

which can be simplified to

$$m_i^{'} = m_i + \delta m \qquad \text{and} \qquad m_j^{'} = m_j - \delta m$$

where δm is defined as

$$\delta m = (1-\lambda)(\epsilon m_j - (1-\epsilon)m_i)$$

To get back the simple model without savings, we can put $\lambda = 0$.That is, if $\lambda = 1$, no transactions are made. In order to save an amount λ it must take the value $0 < \lambda < 1$.

2.3 Nearest neighbor interaction and former transactions

These transactions happen between agents which have preferences for whom they make transactions with. This means not every transaction is accepted like the models from section 2.1 and 2.2. If two agents are financially close, they are more likely to interact [4]. To add this feature there is a need for a probability that the agents i, j will interact, given by

$$p_{ij} \propto |m_i - m_j|^{-\alpha} \tag{2}$$

with $\alpha > 0$, where we get the simple model without the probability for $\alpha = 0$. The tail of the wealth distribution will have a power law form when $\alpha \gg 1$.
There is also a likelihood that two agents that have conducted transactions between them before will do so again. The addition of this feature is done by multiplying p_{ij} with $(c_{ij}+1)^{\gamma}$ which yields

$$p_{ij} \propto |m_i - m_j|^{-\alpha}(c_{ij}+1)^{\gamma} \tag{3}$$

where the number of previous interactions between the two agents is represented by the matrix c_{ij}. To ensure that two agents that have not interacted before can do so, the factor 1 is added to c_{ij}. To get back the model for the financially close interactions without the additional term, we can simply put $\gamma = 0$.

3 Methods and algorithms

3.1 The Monte Carlo scheme

Due to the probabilistic nature of this problem we need to implement an algorithm accordingly. The Monte Carlo scheme is an algorithm that bases it's calculations on random number generators, which is what we use. A flowchart of the Monte Carlo model is illustrated in figure 1[1]. In addition, the implementation is given in listing 1. This shows the basic algorithm, although there are obviously more declarations of the random numbers, agents and so forth. We only use the upper triangular part of the matrix c_{ij} to keep track of agents interactions since $(i, j) = (j, i)$ are interactions between the same agents. The probability from section 2.3 is written as p and tested against a random number r, which makes our sampling condition. We also have an insurance that if we pick $i = j$, there is no money exchange (not that it would make any difference) since it would be the same agent.

```
  void MonteCarloSimulation(Vectors and variables) {
    //Setting up random number generator and declaring variables//
    for (int run = 0; run < runs; run++) {
        //Setting equal amount of money to agents and initialize the
            exchange tracker//
        for (int interaction = 0; interaction < transactions;
            interaction++) {
            //Pick random agents to make a transaction//
            if (i != j) {
                //Pick random numbers epsilon and r
                (i > j) ? c = exchangeTracker.at(j).at(i) : c =
                    exchangeTracker.at(i).at(j);
                if ((agents.at(i) - agents.at(j)) == 0) {
                    p = 1.0;
                } else {
                    p = pow(fabs(agents.at(i)-agents.at(j)), -alpha)*
                        pow((c+1), gamma);
                }
                if (r < p) moneyExchange(agents, exchangeTracker, i, j,
                    epsilon, lambda, exchangeCounter);
            }
            //Computing the variance for run = 1//
        }
        //Stack the money or add result to bins//
    }
}
```

Listing 1: Simulating transactions between agents using Monte Carlo method.

[1]We have to admit that the flowchart is a bit messy.

In the flowchart, figure 1, we added some elements that we stripped from listing 1.

For section 2.1 and 2.2, to get decent statistical data we do 10^4 independent runs of the simulation. After each run, with a given number of transactions, the money from the agents is sorted in descending order and stacked in the money vector m. Before writing to file, the data is normed by the number of runs, making the data an average over these runs. The data from these sections was made into histograms using Python. For section 2.3 we had difficulties reproducing the same plots as in [4], and decided to do a little rewrite of the program to produce the histogram in a vector with bins representing different incomes. To save some runtime we do 10^3 runs with 10^6 transactions.

In all our cases we have put the initial amount of money to $m_0 = 1$ such that $\sum_{i=1}^{N} m_i = N$ giving

$$\langle m \rangle = \frac{1}{N}\sum_{i=1}^{N} m_i = \frac{1}{N}\cdot N = 1 = m_0 \qquad \text{and} \qquad \beta = \frac{1}{\langle m \rangle} = 1$$

Changing the number of agents, N, does not change the initial amount, m_0, given to each agent and $\langle m \rangle$ will be equal to m_0. The rest of the parameters is specified in the captions of the figures.

The variance of a set of N equally likely values we calculate by

$$\sigma^2 = \mathrm{Var}(m) = \frac{1}{N}\sum_{i=1}^{N}(m_i - \mu)^2 \qquad \text{where} \qquad \mu = \frac{1}{N}\sum_{i=1}^{N} m_i = m_0$$

Knowing that the expected value $\mu = m_0$ tells us that the variance will fluctuate around the initial amount of money m_0 given to each agent.

In listing 2 we have added the code where we calculate the variance for run $= 0$. Each run is independent, meaning that the variance should look similar for each run and fluctuate around m_0.

```
if (run == 0) {
    double var = 0;
    for (int k = 0; k < N; k++) {
        var += (agents.at(k) - m0)*(agents.at(k) - m0);
    }
    double variance = var/N;
}
```

Listing 2: Computing the variance.

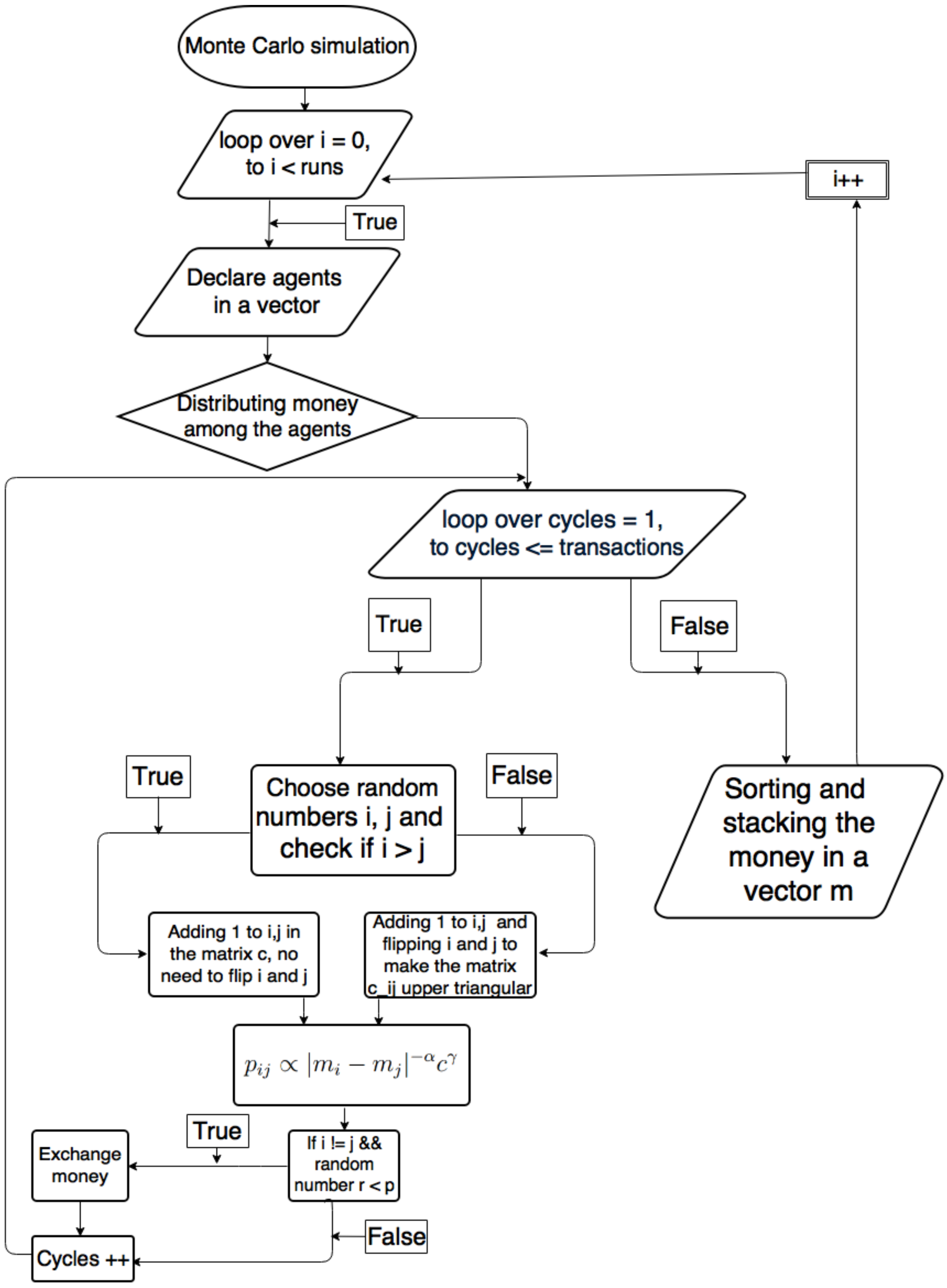
Monte Carlo simulation
loop over i = 0, to i < runs
i++
True
Declare agents in a vector
Distributing money among the agents
loop over cycles = 1, to cycles <= transactions
True
False
True
Choose random numbers i, j and check if i > j
False
Sorting and stacking the money in a vector m
Adding 1 to i,j in the matrix c, no need to flip i and j
Adding 1 to i,j and flipping i and j to make the matrix c_ij upper triangular
$p_{ij} \propto |m_i - m_j|^{-\alpha} c^{\gamma}$
True
Exchange money
If i != j && random number r < p
False
Cycles ++

Figure 1: Illustration of the Monte Carlo algorithm. The basic idea is to initialize some variables, generate random numbers, generate paths and calculate the averages of your choice.

3.2 Runtimes

Monte Carlo methods can be time consuming when running for large numbers, but it is not the only thing that can be time consuming. We have picked out a few selected runtimes to show. In table 1 we can see that using $\alpha = \{0.5, 1.5\}$ makes the runtime about three to four times as long compared to $\alpha = \{1.0, 2.0\}$. The C++ power function *pow*() can take real numbers (and complex numbers) as exponent, but when the number can't be converted to an integer it will take extra time to calculate the result. Using more agents, transactions and more runs also give a longer runtime, but that is kind of obvious. Changing the factors λ, α and γ does not change the runtime by much as long as they can be converted to integers by the *pow*() function.

N	Transactions	Runs	λ	α	γ	Runtime [s]
500	10^6	10^3	0	0.5	0	328
500	10^7	10^3	0	0.5	0	3475
500	10^6	10^3	0	1.0	0	88
500	10^6	10^3	0	1.5	0	319
500	10^7	10^3	0	1.5	0	3374
500	10^6	10^3	0	2.0	0	89
500	10^7	10^3	0	2.0	0	1074
1000	10^6	10^3	0.9	2.0	4.0	123
1000	10^7	10^3	0	1.0	0	1100
1000	10^7	10^3	0.9	2.0	0	1103
1000	10^7	10^3	0.9	2.0	4.0	1218
1000	10^7	10^4	0	1.0	0	7812
1000	10^7	10^4	0	1.5	0	30900
1000	10^7	10^4	0.25	1.5	0	31151
1000	10^7	10^4	0.25	2.0	0	7898

Table 1: Runtimes given in seconds for different parameters. Each row gives a set of parameters and the runtime for those parameters are in the last column. The runtimes are just an indication (one sample) of how long the program takes to run. If we wanted a more exact runtime, we would have run the program at least 10 times for each set of parameters and taken the average of these.

If we had implemented a condition to find the steady state when equilibrium is reached we could probably have saved a lot of runtime. When the system reaches equilibrium there is just minor changes in the transactions. It would not have given us any greater insight.

4 Results and discussion

4.1 Agent based simulation and the exponential distribution

Running the program for the simple transaction model based on the random number, ϵ, we got the histogram shown in figure 2. This result is quite striking when it comes to the inequality of the distribution of wealth. There is a lot of agents that are left with very little money, and few with a lot.

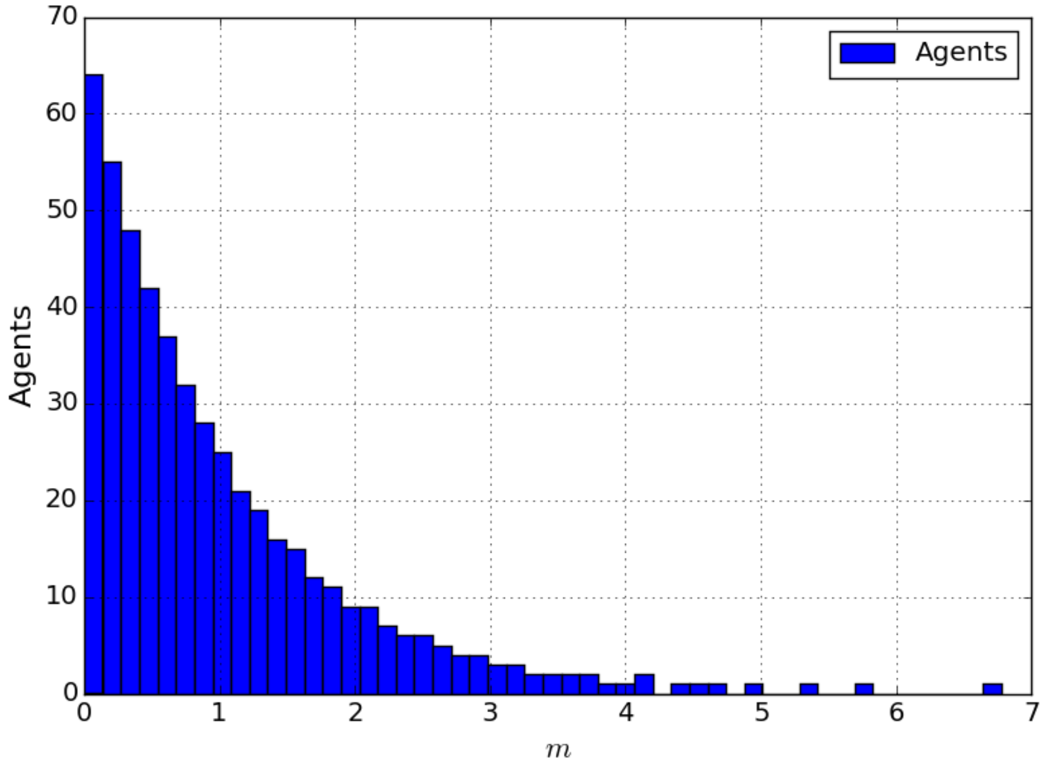

Figure 2: [Parameters: N = 500, $m_0 = 1$, transactions = 10^7, runs = 10^4]. Distribution of wealth after a simulation of agent based transactions between two random agents at each step. A vital point is that this model is stripped of realistic aspects of actual human trade as of now.

To work with statistical analysis it is of interest to change units. By normalizing the histogram in figure 2 we get the probability density, $P(m)$, as shown in figure 3. The wealth distribution given by the Gibbs distribution, $\omega_m = \beta e^{-\beta m}$, is fitted to the normalized histogram [3].

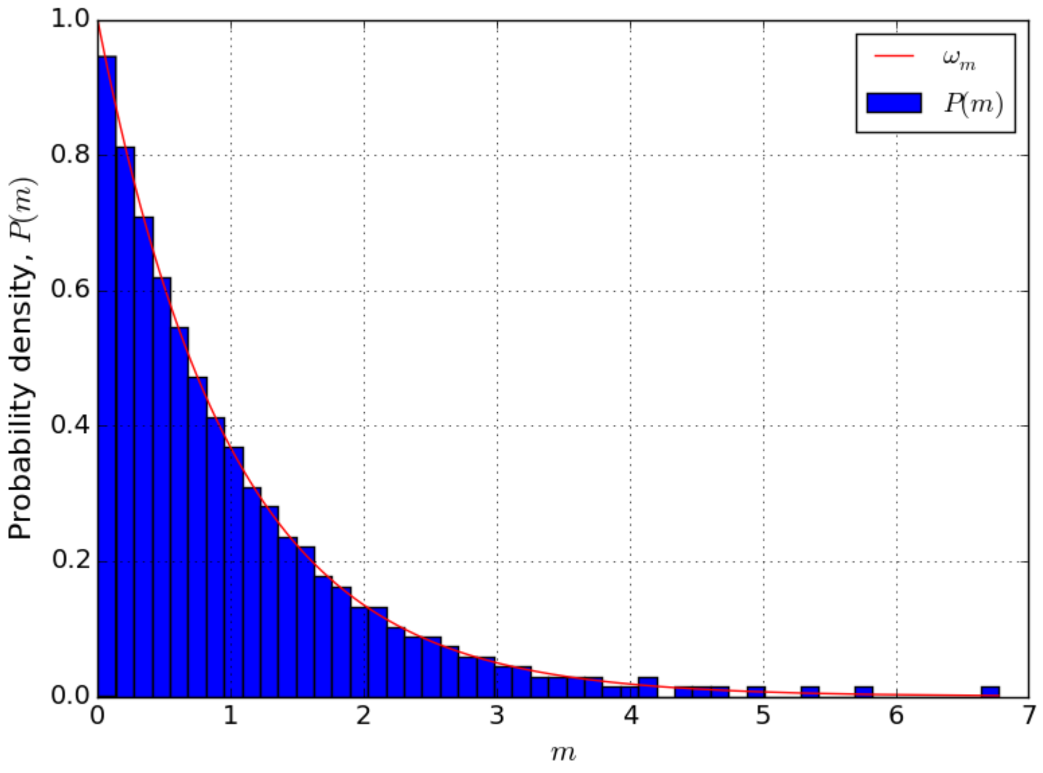

Figure 3: Figure 2 normalized and fitted with the Gibbs distribution ω_m.

The Gibbs distribution is one of the key results of the simulation of the transactions without using any of the parameters such as λ for saving some of the money. This is illustrated in figure 4 where the straight line implies that the distribution follows an exponential curve. That is, this is what we would expect out of the logarithm of equation (1). What this shows is that our program functions as it should.

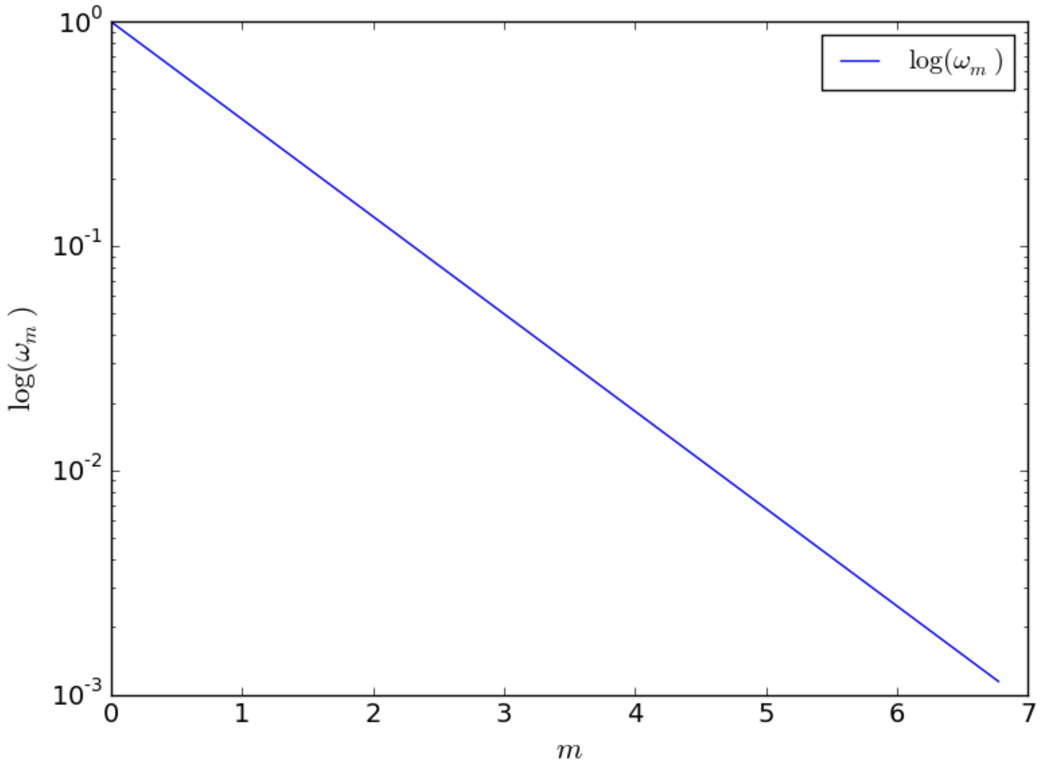

Figure 4: Plot of the logarithm of the Gibbs distribution ω_m.

Figure 5 shows the variance for the simple model. This seems to be correct from what we would expect from section 3.1. We can see that the variance fluctuates around $m_0 = 1$. The variance seems to have reached some sort of equilibrium after about 10000 to 20000 transactions.

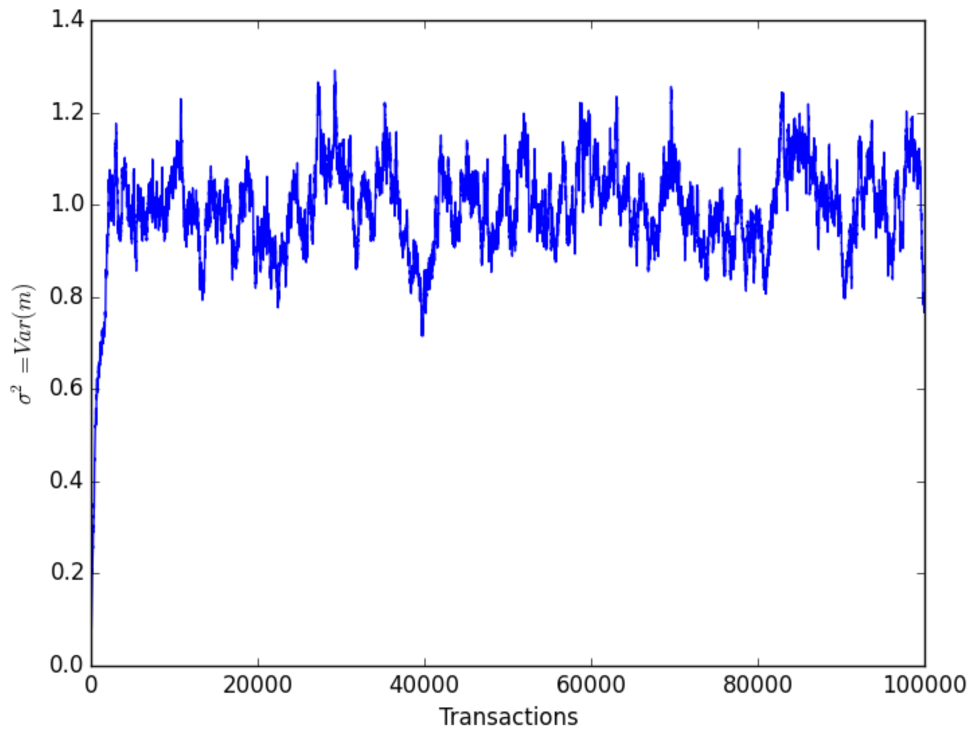

Figure 5: The variance of the money m as a function of the number of transactions for the simple model.

To see if we have reached equilibrium at 20000 transactions we have plotted the absolute difference between 10^5 transactions and 20000 transactions in figure 6. We can see by the y-axis that there is not much difference between the two distributions, meaning we have reached some kind of equilibrium after 20000 transactions for the simple model. This will not be the case when we introduce the variables λ, α and γ later on.

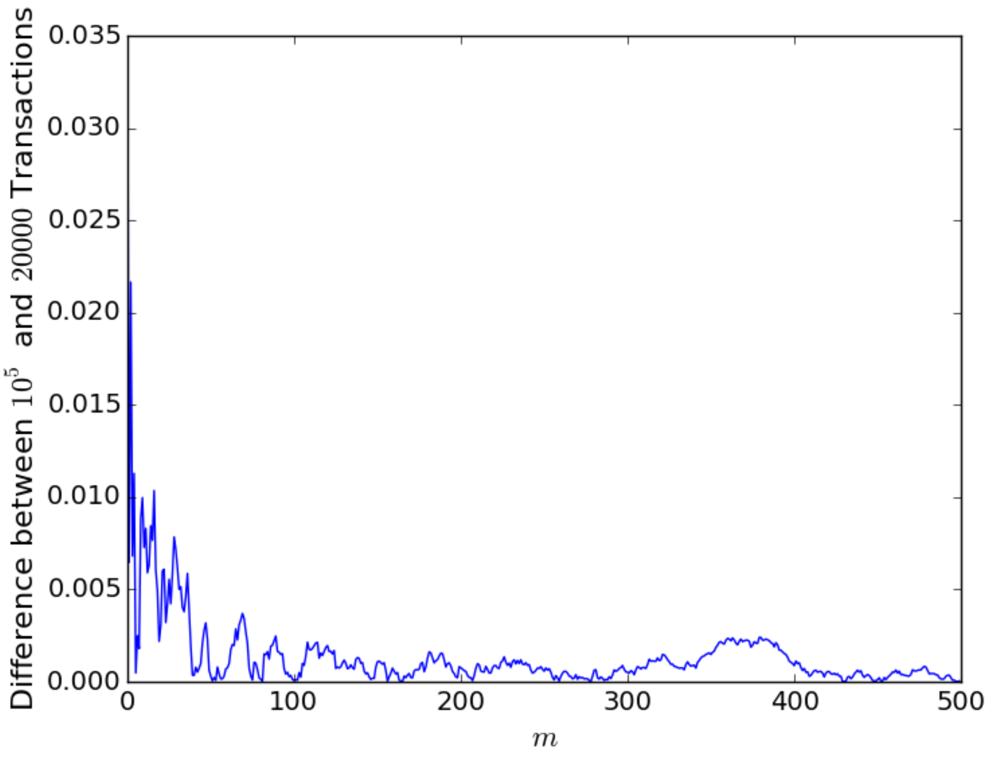

Figure 6: The difference between the variance for 10^5 transactions and 20000 transactions for the simple model.

4.2 Savings and the Gibbs distribution

In this model of a closed economy we have simulated the transactions between pairs of random agents (i, j), ultimately giving us the distribution of wealth after a given number of runs. The parameter λ characterizing the savings of the agents play a vital role when it comes to the form of the probability distributions. In figure 7 we can see that we get the well known Gibbs distribution for $\lambda = 0$, but when $\lambda \neq 0$ we get a different distribution which tends towards a normalized curve for the upper range of $\lambda \in (0, 1)$. This means that the empirical distributions does not follow the ones observed when the saving parameter is added, and we need another kind of distribution.

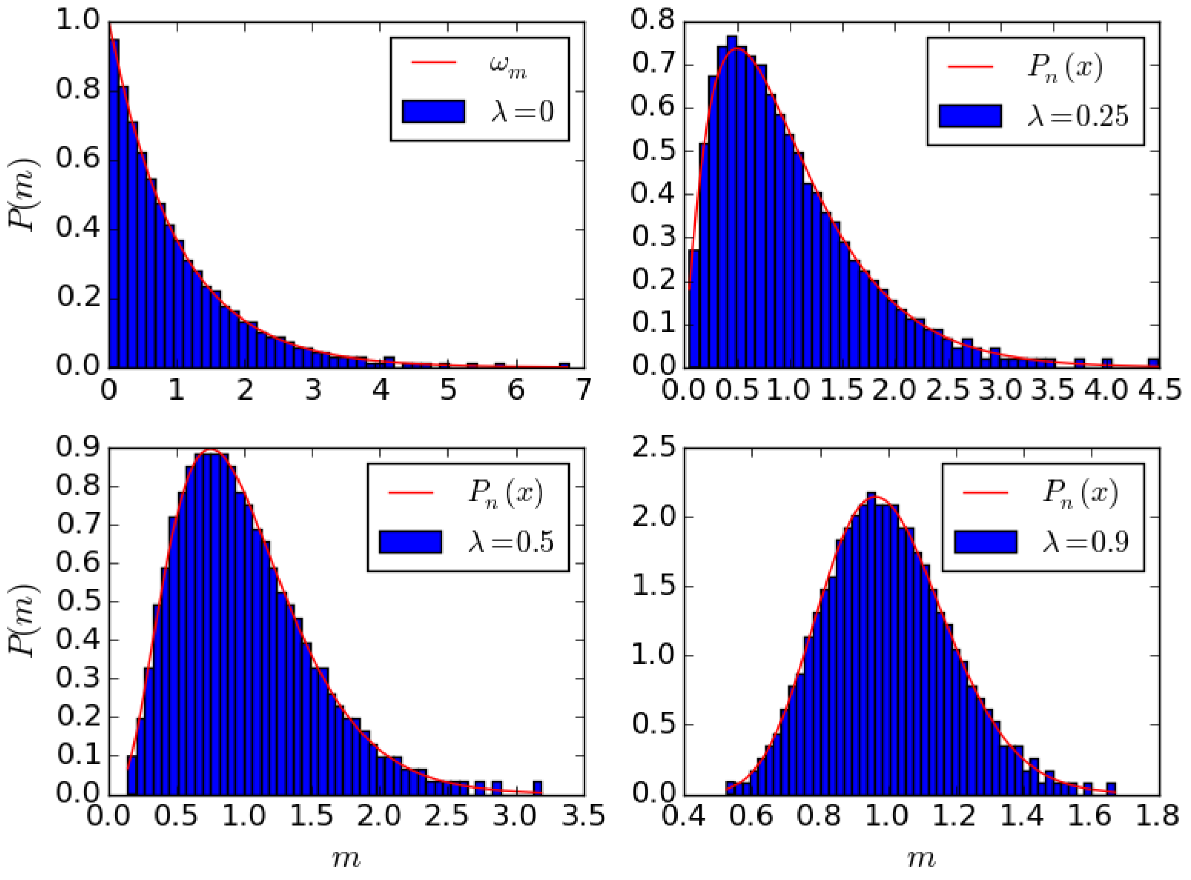

Figure 7: [Parameters: $N = 500$, $m_0 = 1$, transactions $= 10^7$, runs $= 10^4$]. Probability density function of the wealth with and without the saving parameter λ. The Gibbs distribution, ω_m, is fitted for $\lambda = 0$ and the distribution from equation (4) is fitted for the other values of λ.

Patriarca, Chakraborti and Kaski [3] found in their article that in the fitting of the model it was convenient to introduce some new variables. The reduced variable

$$x = \frac{m}{\langle m \rangle},$$

which is the agents money in units of the average money $\langle m \rangle$, and the parameter

$$n(\lambda) = 1 + \frac{3\lambda}{1-\lambda}.$$

They found the function to fit the wealth distributions to be

$$P_n(x) = a_n x^{n-1} e^{-nx} \tag{4}$$

where x and n are defined above, and the normalization condition gives

$$a_n = \frac{n^n}{\Gamma(n)}$$

where $\Gamma(n)$ is the Gamma function. The distribution function from equation (4) is used in figure 7 for $0 < \lambda < 1$ in order to parametrize the probability densities. As we can see, they are in very good agreement with each other.

The new distribution function, $P_n(x)$, still has an exponential factor e^{-nx} similar to the Gibbs distribution. The average value is rescaled by n, but it is the power x^{n-1} that changes the shape of the distribution. Further [3] by introducing the rescaled variable

$$x_n = nx \equiv \frac{m}{\langle m \rangle / n}$$

and the corresponding probability density

$$P_n(x_n) = \frac{dF_n(x)}{dx_n} \equiv \frac{P_n(x)}{n},$$

where F_n is the cumulative function, and by using the explicit expression of a_n, the distribution from equation (4) becomes

$$P_n(x_n) = \frac{x_n^{n-1} e^{-x_n}}{\Gamma(n)} \equiv \gamma_n(x_n) \tag{5}$$

which is the γ-distribution function, $\gamma_n(x_n)$, for the variable x_n.

The relation between the curves in figure 7 is shown in figure 8 and 9. Due to the exponential tail of the Gibbs distribution, there is a majority of poor agents and a few rich agents. By including the saving parameter λ and let $\lambda \to 1$, we see that the wealth distribution tends to be more normalized around 1. This means that there are a higher probability of having an average income, rather than having next to nothing or having a lot. We cannot put λ to 1, because then no one would interact with each other. The figures 8 and 9 has the same form as the figures fig. 1 and fig. 2 in [3].

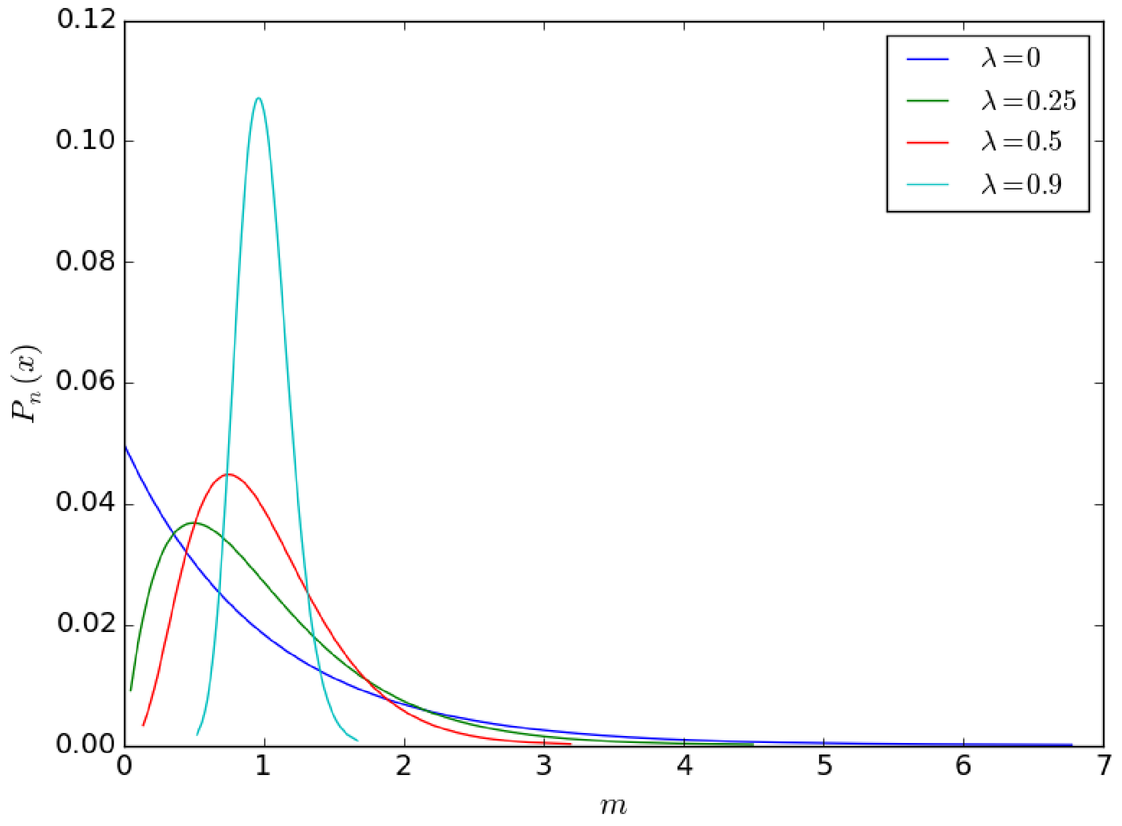

Figure 8: Probability density functions as shown in figure 7. As $\lambda \to 1$ the distribution is more normalized around 1 and no longer follows a Gibbs distribution.

The relaxation of the system towards the equilibrium wealth distribution is invariant irrespective of the initial distribution of wealth.

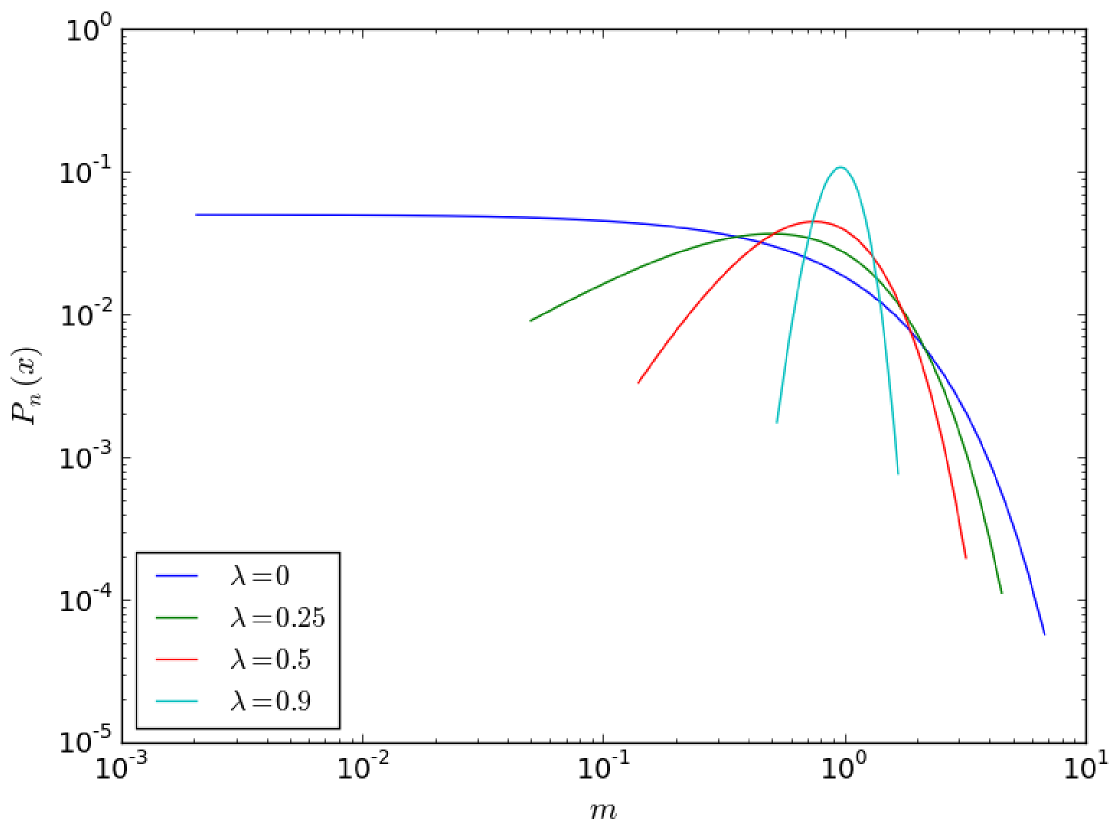

Figure 9: Logarithmic plot of the probability distribution from figure 8.

In figure 10 we extract the high-end tails of the distributions to see if they follow Pareto's law. As we can see they do not follow Pareto's law since the Pareto exponent $\alpha \notin [1, 2]$. They have the values of α between 4.5 and 13 based on the saving parameter λ. They do at least follow a power law.

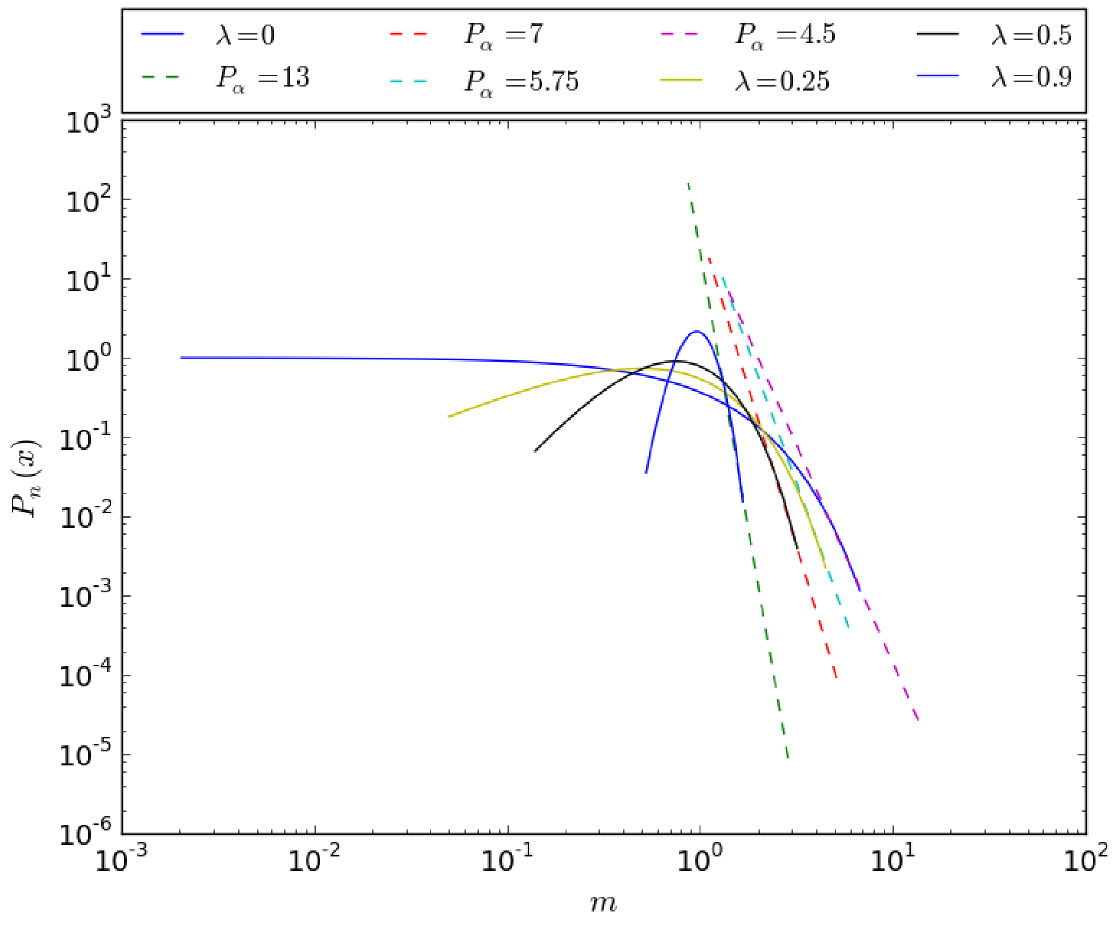

Figure 10: Parametrization of the high-end tails of the distribution on terms of power laws. P_α represents the exponent α in $m^{-1-\alpha}$.

4.3 Nearest neighbor interaction and former transactions

The criteria for the model where $p_{ij} \propto |m_i - m_j|^{-\alpha}$ is that they are close financially, and so depending on α we get different results. The extremes returns the Gibbs distribution and the tail of the wealth distribution follows a power law. That is, for $\alpha = 0$ we get the Gibbs distribution and for $\alpha \gg 1$, the tail of the distribution should have the form of a power law. The results we get for this probability is shown in figure 11. The figure does not look exactly like the fig. 1 in [4]. It is not easy to know what they have used for normalizing the data and how they have plotted it, because the information about the process is vague.

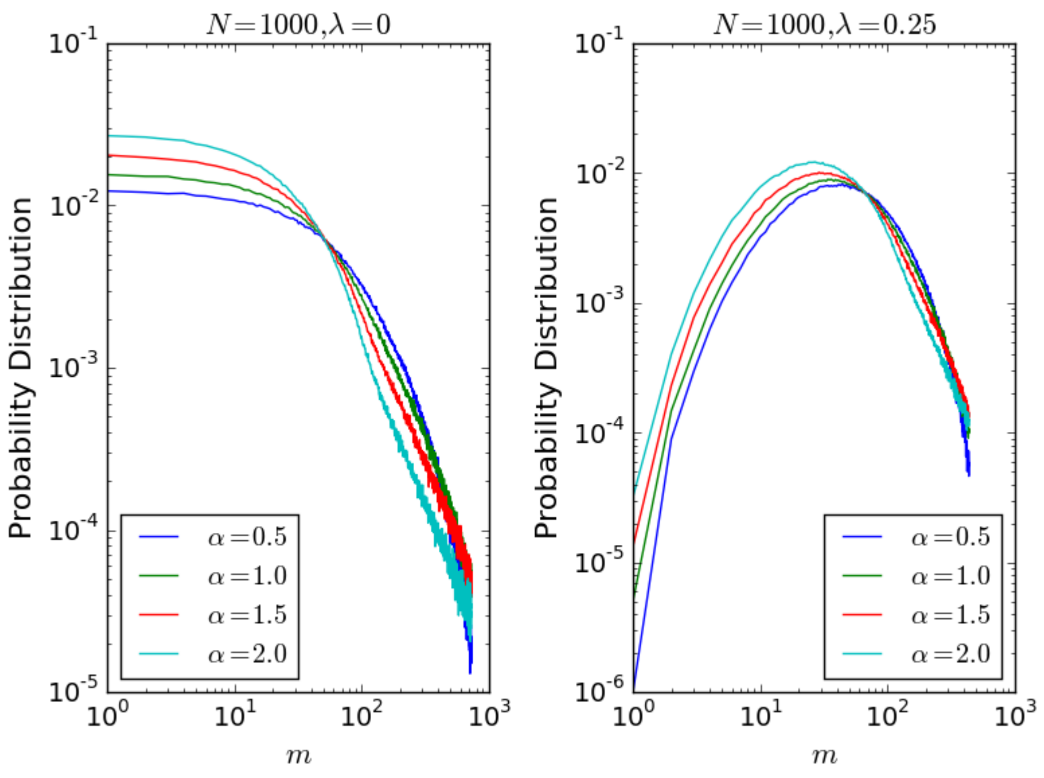

Figure 11: [Parameters: transactions = 10^7, runs = 10^3]. The probability distribution with and without saving, λ, using the constraint from equation (2).

Here we have chosen the figure for $N = 1000$ agents. The plot for $N = 500$ is almost identical. The curves for a higher number of agents are a bit smoother. This clearly shows a power law behavior for the richest people. There is also a spread in the functions for different α's.

We can see from figure 11 that the probability of having a lot of money when there is no saving involved is low, and the probability of having next to nothing is high. With the element of saving, there is a higher probability of having an average of money which means there is a low probability for having next to nothing or to have a lot. A higher α shifts the probability for having less money to be more probable.

The power-law tail can be parametrized by the lines as seen in figure 12. Some of the curves follows Pareto's law, but not all of them.

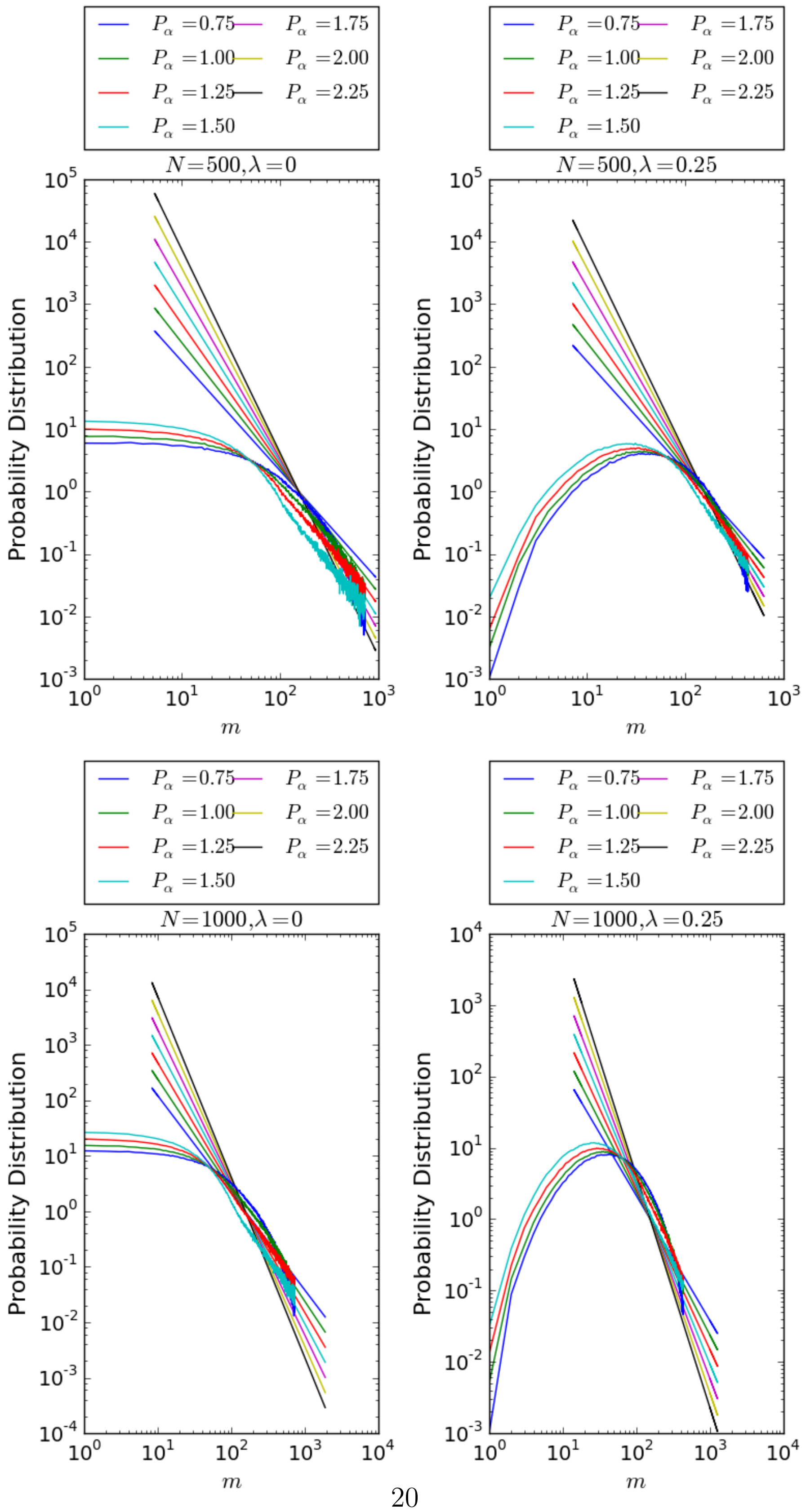

Figure 12: The probability distributions as seen in figure 11 with the high-end tail parametrizations. P_α represents the exponent α in $m^{-1-\alpha}$.

It becomes apparent why it's called a power-law tail. There's just a small fraction of the probability distribution that follows the parametrizations. To illustrate this further we've plotted the 40% richest agents of the distribution in figure 13.

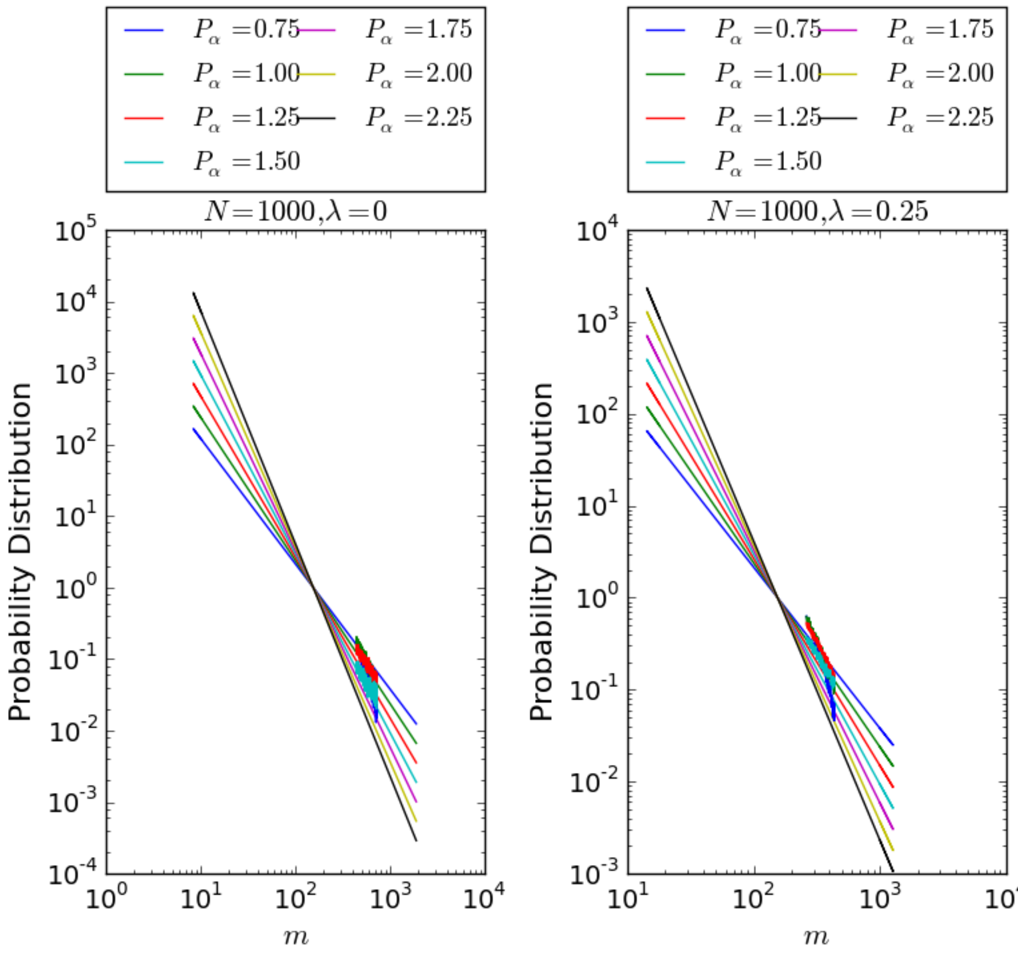

Figure 13: When we cut 60% of the probability distribution we're left with the part that follows the power-law parametrizations. In other words, this plot contains the top 40 % with respect to income.

Figure 13 shows the 40 % richest in the different distributions. If we had known how to do linear regression in Python, we could have done this in a better way. A better way to find the high-end tails is to only plot the 10-15 % richest agents and find a line to fit using linear regression analysis. In that way we could have found one line for each set of parameters. The way we have done it is in the 'spray and pray' kind of way.

We would expect to get a higher value for the parameter in Pareto's power law with higher α, but our plots does not show how this works. Our program works as expected up until we add the nearest neighbor interaction. There might be something wrong with the way we calculate the probability. One of the fundamental properties of a probability in general is that $\sum_i p_i = 1$, but if the values for p are off this might not be the case.

The variance seems to be correct, but the variance and the mean etc. of a probability distribution function (PDF) can be correct while still not getting the desired form of the PDF itself. This is due to the specific quantitative measure of a set of points called moments [5]. The zeroth moment of a probability distribution is it's total probability, the first is the mean value, and the second is the variance. The collection of all these moments of all orders from $0 \to \infty$ represents the entire probability distribution.

It is possible that the lack of restrictions in our program gives these results. Our program does not stop when it reaches the steady state, and thereby it is possible that the variance can fluctuate more after a while or even run off in one direction. By restricting the program to stop when it's in the steady state, it will stop at different transactions for different parameters. This restriction could have been added by looking at how the variance of the variance fluctuates and giving it a maximum percentage of fluctuation to tell if it is in the steady state.

This seems like a major problem, and it probably is, but the point of this study is to show that we can in fact model complex systems like the stock market, which consists of interacting agents with some simplifications by the use of statistics. The trick is to model these systems as a whole, that is, as a large system analogous to that of a box filled with an ideal gas where the all the microstates make up the macrostate while still taking the individual interactions into account as far as collisions go.[2].

In the paper written by Michal Brzezinski [6] they found that the wealth distribution which was taken from the richest persons in the world published by Forbes and other such magazines followed a power-law behavior on average no more than 60% of the largest wealth values. This is interesting because if our data was correct we would conduct the same study on the wealthy people of Norway and compare the supposed power-law tails to the results given by Michal Brzezinski [6].

[2]We've written a small appendix on the analogy between the kinetic theory of the ideal gas and wealth distributions. There is a lot more to be said on this subject, in fact, there is something called: *The thermodynamics of money dynamics.*

The transactions between the agents are made more realistic when we add a feature where the agents are more conservative when it comes to who they interact with. That is, they are more likely to interact with agents they have interacted with previously. This effect is controlled by the parameter γ. We can add the feature for the former transactions with $p_{ij} \propto |m_i - m_j|^{-\alpha}(c_{ij}+1)^{\gamma}$, and we have tried changing λ, α and γ for 1000 agents to see what differences they make. Figure 14 shows a similarity to the nearest neighbors interactions without former transaction preferences.

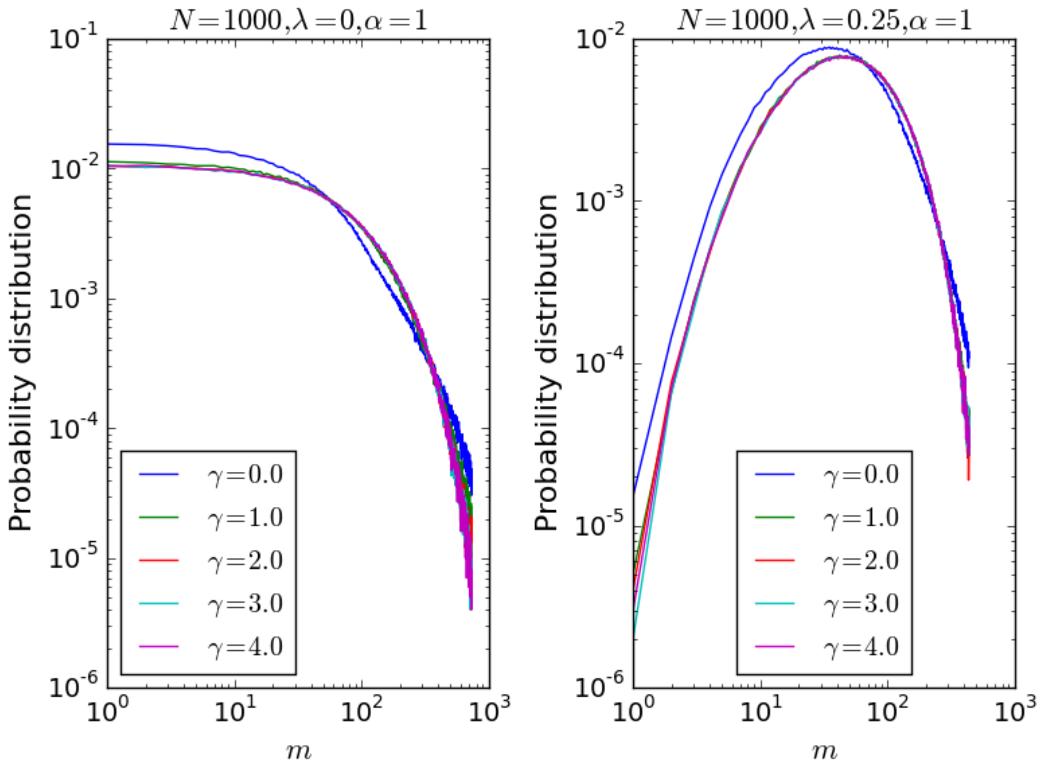

Figure 14: [Parameters: transactions = 10^6, runs = 10^3]. The probability distribution with and without saving, λ, using the constraint from equation (3).

The parameter γ does not seem to do much difference for the curves, and if we compare figure 14 and figure 15 we can see that α is a parameter giving the curves a spread.

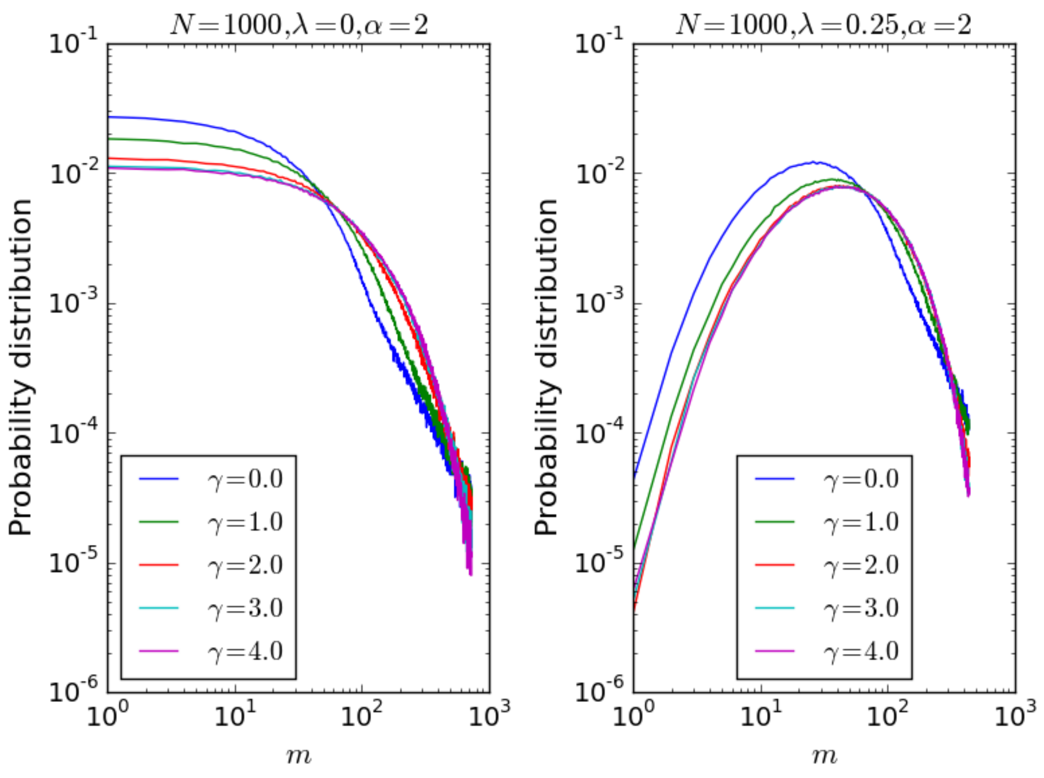

Figure 15: [Parameters: transactions = 10^6, runs = 10^3]. The probability distribution with and without saving, λ, using the constraint from equation (3).

The plots and results in the article by Goswami and Sen[4] are quite hard to reproduce. For example, the back end of their distributions in fig. 5[4] is something that we could not reproduce as shown in figure 14 or 15. However, the power-law tail is clearly shown in those figures and they can be parametrized by a linear function. We have done so in figure 16. Unfortunately we have used the same approach finding the lines as we did with the nearest neighbor model. It is not easy to see which curves follows Pareto's law, but some of them do.

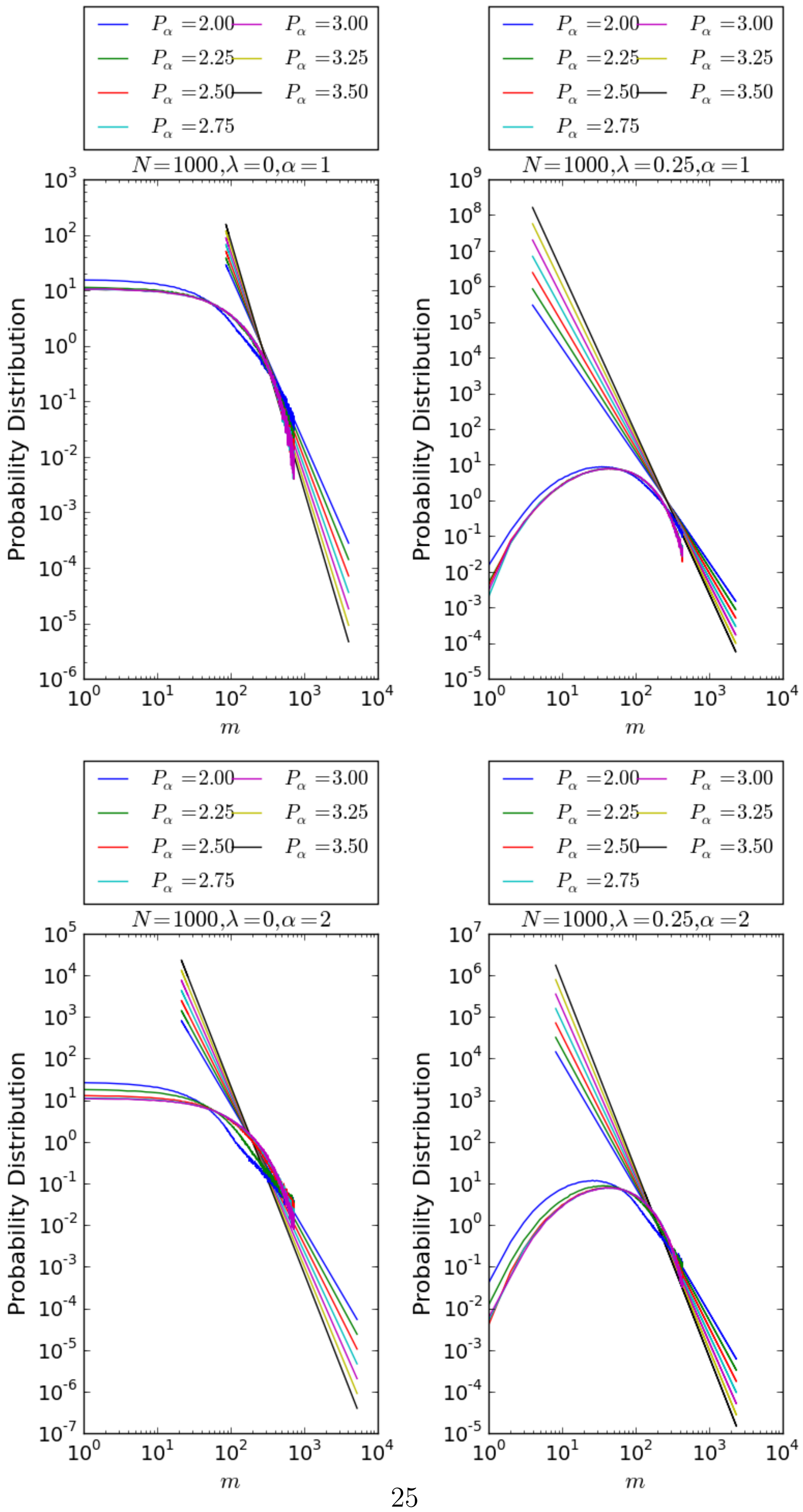

Figure 16: The probability distributions as seen in figures 14 and 15 with the high-end tail parametrizations. P_α represents the exponent α in $m^{-1-\alpha}$.

The parametrization for $\alpha = 2$ and $\lambda = 0.25$ seems to follow the parametrization better than the corresponding plot for $\alpha = 1$. The common denominator here, as with the previous case, is that the higher end of the income follows this power-law. We missed with the parametrization of $\lambda = 0.25$ and $\alpha = 1$. The slope is even steeper, meaning that P_α should have been larger.

4.4 Additional features of the agent based model

There are of course more sides to a human being than what we have presented in this paper, and if we could add more of those we would make the model more realistic. We could for example add some form of curiosity, exploratory behavior and an ability for innovation. If we were to add these aspects they would have to be very simplified compared to the real abilities in humans. A large problem, or rather the main problem with this would of course be how one would replicate these aspects mathematically. A learning and teaching ability would also help to make this more realistic. As the agents trade, they also get more experienced and get to know how to 'play the game' so to speak. The agents would surely try to predict the outcome of future transactions in order to find the best transaction for themselves. This implies that a feature where the agents have future expectations would also increase the realism of the model.

5 Conclusion

We have conducted a study of the distribution of wealth in a closed economy among agents that start out with an initial amount of money m_0. The simulation is made more realistic with the addition of α which tells us that the agents are far more likely to interact with other agents that are closely related to them in the wealth space when $\alpha > 0$. Other parameters to imitate a realistic situation was also added. γ represents the previous interaction of the agents, which means that they are more likely to interact if they have done so before, and β represents the selection of agents with a probability proportional to their wealth. The savings the agents make before a transaction was modeled by the parameter λ which showed that when $\lambda \to 1$, the distribution looks like a Gaussian, rather than a Gibbs distribution.

The higher end of the tail in the probability distributions with and without the memory of the agents shows the same thing; they both follow a power-law. This indicates that even though we add more realistic features to the model, the power-law tail can still be seen in the probability distribution.

We found that we could reproduce some of the results of Goshwami and Sen[4], but the full results were not reproduced as we discussed earlier. This might be due to the fact that there is little information about the process where they found the results and little discussion of importance of what the results mean. One of the key things in this study is that the distribution of wealth between the agents is predictably unbalanced.

To capture physical and human behavioral features of economic systems while still retaining analytical agreement is very difficult through a classical mathematical manner. The agent based modeling has become popular among econophysicists and is a step toward a more complete model of the economic systems of the world. One of the great advantages of this is that it opens up the possibility for modeling these systems with a vast number of agents which will generalize the studies of the economic systems.[3]

6 Appendix

6.1 The relation between the kinetic theory of the Ideal gas and wealth distributions

Benoit Mandelbrot once said: " *There is a great temptation to consider the exchanges of money which occur in economic interaction as analogous to the exchanges of energy which occur in physical shocks between molecules. In the loosest possible terms, both kinds of interactions should lead to similar states of equilibrium. That is, one should be able to explain the law of income distribution by a model similar to that used in statistical thermodynamics: many authors have done so explicitly, and all the others of whom we know have done so implicitly*" [7]. When two particles in an ideal gas collide they exchange kinetic energy depending on the angle of the collision and the dimension of the system. This is analogous to the exchange of money between two randomly selected agents in our model. Before we add any features such as agent (i, j)'s exchange history the dynamics follow a Markovian process given by

$$\begin{pmatrix} m_i(t+1) \\ m_j(t+1) \end{pmatrix} = \mathcal{M} \begin{pmatrix} m_i(t) \\ m_j(t) \end{pmatrix}$$

where $m_j(t)$ is agent j's income at time t and $\mathcal{M}$ defines the exchange process. The money in each transaction is conserved and no agent ends up with debt. This process is illustrated in figure 17.

[3]And we all know that physicists think in terms of the general aspects of systems!

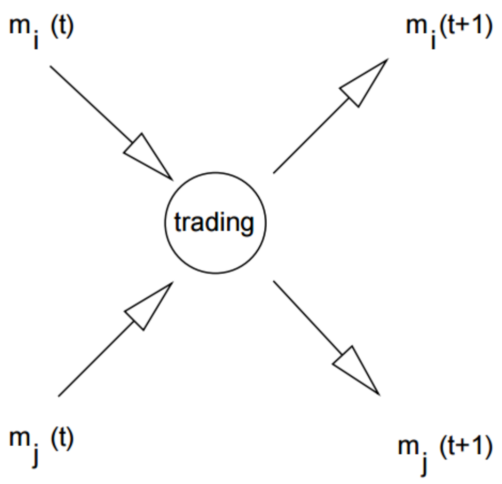

Figure 17: Diagram illustrating the trading process between agent i and j with an initial amount of money m_i and m_j.

Consider an ideal fluid in one dimension. The particles can only collide head on, and this would resemble the situation where the agents exchange money without saving any of it beforehand. However, if we increase the dimension of the fluid system the average of the collisions would only exchange a fraction of the kinetic energy due to the different angles between the particles upon collision, that is $\sim 1/D$ where D is the dimension of the system. Thus, the saving parameter λ is analogous to $\lambda \sim 1 - 1/D$. That is, the act of saving some of the money represents the fact that not all kinetic energy is transferred during the collision.

6.2 Pareto's principle

In 1941, the Romanian-born American engineer and management consultant Joseph M. Juran discovered what he christened Pareto's principle which essentially states that 80% of a problem is caused by 20% of the causes. This principle is also known as *the vital few and the trivial many* and *the vital few and the useful many*. The latter was preferred by Juran in his later years. This principle is applied in a vast variety of fields, including computer science and business. The interesting thing about this principle is that it follows a power law distribution, or the Pareto distribution for a set of parameters. There's actually many phenomena in nature that exhibit this kind of behavior as well which implies a deeper consequence of the principle. These phenomena range from earthquakes to solar flares and man-made phenomena as well as discussed in M. E. J. Newman's article [8].

A graphical description[9] of the principle is shown in figure 18 below.

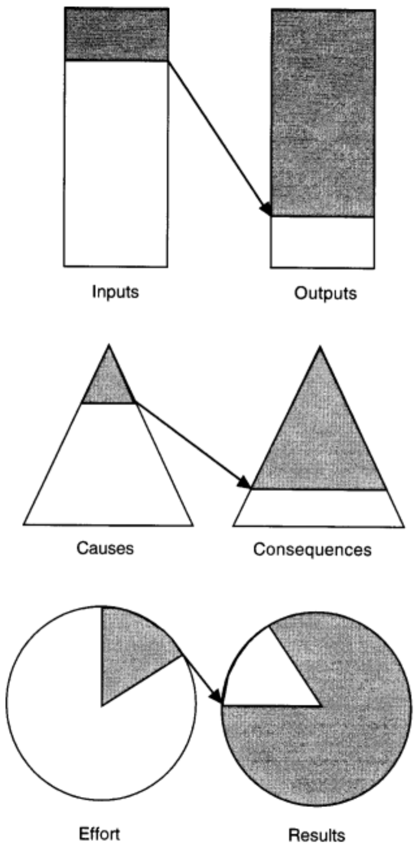

Figure 18: Illustration of the Pareto principle. 20% of the input generates 80% of the output and so forth.

In the 1960's and 1970's IBM made use of the 80/20 as one of the earliest and most successful corporations, and in 1963 they discovered that most of the time spent by the computers was used executing about 20 percent of the code. In fact, they found that the amount of time spent was about 80%. This lead to the immediate rewriting of the software to make the the 20 % very user friendly and accessible. To the great joy and triumph of IBM and the frustration of their competitors, this made IBM's computers much more efficient and faster for the majority of applications. All manufacturers of personal computers in the coming decades made use of this principle, including Apple, Microsoft and Lotus. They even did it with more fervor to make the computers easier to use and cheaper for the incoming horde of customers[9].

References

[1] V. Pareto. Cours d'economie politique. Rouge, Lausanne and Paris, 1897.

[2] Morten Hjort-Jensen. Computational physics, lecture notes fall 2015. Department of Physics, University of Oslo, 2015. https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[3] Marco Patriarca, Anirban Chakraborti, and Kimmo Kaski. Gibbs versus non-gibbs distributions in money dynamics, 2004. ISSN 0378-4371. URL http://www.sciencedirect.com/science/article/pii/S0378437104004327.

[4] Sanchari Goswami and Parongama Sen. Agent based models for wealth distribution with preference in interaction, 2014. ISSN 0378-4371. URL http://www.sciencedirect.com/science/article/pii/S0378437114006967.

[5] Sample page from Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Recipes Software. http://www.aip.de/groups/soe/local/numres/bookcpdf/c14-1.pdf.

[6] Michal Brzezinski. Do wealth distributions follow power laws? evidence from 'rich lists'. Faculty of Economic Sciences, University of Warsaw, Dluga 44/50, 00-241, Warsaw, Poland.

[7] Arnab Chatterjee and Bikas Chakrabarti. Kinetic exchange models for income and wealth distributions. Theoretical Condensed Matter Physics Division and Centre for Applied Mathematics and Computational Science, Saha Institute of Nuclear Physics, 1/AF Bidhannagar, Kolkata 700064, India. Economic Research Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata 700108, India.

[8] M. E. J. Newman. Power laws, pareto distributions and zipf's law. Department of Physics and Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109. U.S.A.

[9] Richard Koch. The 80/20 principle. Nicholas Brealey Publishing Limited, ISBN 1-85788-167-2 HB, London 1998. leadershipcoachingblog.com/wp-content/uploads/2012/03/the-80-20-principle-to-achieve-more-with-less-effort1.pdf.