Modeling a financial market numerically using Monte Carlo methods

Anders Bråte and Elin Finstad
Institue for Physics
University of Oslo
Norway
December 19, 2019

In modern finance, the modeling and simulation of complex financial systems has become extremely important in predicting the outcome of investments and similar economic markers. The fact that an incredibly large amount of variables affect systems like the stock market, make predicting these practically impossible with a deterministic, or even analytical equation. We therefore resort to a Monte Carlo method, with a fairly simple model consisting of agents trading pseudo random amounts of money at different rates. We consider factors such as previous transactions, saving and proximity in wealth in order to get a more realistic model. Our model and algorithms were able to produce a satisfactory simulation of a closed financial system. We were able to confirm behaviours which are also observed in empirical studies of real life financial systems, such as the power law for the higher tails, and Gibbs distribution for the majority of the agents once the saving, nearest neighbours and former transactions methods were implemented, as expected.

I. INTRODUCTION

in addition the method and algorithms as a whole were able to provide a satisfactory simulation of a closed financial system. We were able to confirm behaviours which are also observed in empirical studies of real life financial systems, such as the power law for the higher tails, and Gibbs distribution for the majority of the agents once the saving, nearest neighbours and former transactions methods were implemented, as expected.

In finance, the modeling of complex systems is often necessary when trying to predict the outcome of an investment, portfolio, holdings or similar economic markets and values. This can be done with certain deterministic calculations. However, the larger the system, and the higher the degrees of freedom, the more of an advantage one will get from using Monte Carlo methods. Variables such as the interest rates for loans, which in large degree affects the spending capability of citizens, elections, which might induce instability in a country's market, availability and abundance of natural resources and many more effects all play into how a given stock moves. It is then apparent that a large number of degrees of freedom, and seemingly random behaviours affect the market, which makes it ideal for modeling with Monte Carlo methods.

In this paper we look at a model of simulating a closed market system, with different parameters and variables which affect the behaviour of the system. Initially we look at the bare bones model of agents exchanging money, which we call the simple model. In order to observe how the system is altered we add a saving model. This is simply a variable λ which denotes a given percentage of money which is saved during a transaction (the saving model). Then we add in a probability which considers the money amounts of the agents to decide whether a transaction will be made (Nearest neighbours and acquaintances). We then add in a factor which weighs whether the agents have done transactions previously or not (Former transactions).

II. THEORY

In our model we will simply look at a certain N number of agents, or stockbrokers if you like, and assume that these exchange money amongst each other. All agents start with the same amount of money, and when two randomly chosen agents exchange money with each other, the amount is given by a random number ϵ following a uniform distribution. For the initial amount of money m_i and m_j for two random agents i and j, the new amount of money is written as

$$m_i' = \epsilon(m_i + m_j). \tag{1}$$

Since money must be conserved ($m_i + m_j = m_i' + m_j'$), the money gained by the one agent, must be subtracted from the other, giving us the equation

$$m_j' = (1 - \epsilon)(m_i + m_j). \tag{2}$$

Here m_i' and m_j' shows the respective agents new money amount.
Due to the nature of the exchanges above, no agent will be left with a debt, meaning m will always be larger than or equal to zero.
Since the amount of money is always conserved, it can be shown that the system reaches an equilibrium state following the Gibbs distribution [7]

$$w_m = \beta exp(-\beta m), \tag{3}$$

where $\beta = \frac{1}{<m>}$, and $<m>= \sum_i m_i/N = m_0$. Here m_0 is the original sum of money each agent had from the start.
In practice, this means that after reaching equilibrium, a small number of agents will be left with the majority of the money, and the rest will be left will a significantly smaller sum.

The Saving Model

In trying to make our model more advanced to better fit reality, we introduce saving. Each agent, once chosen to make a transaction will save a bit of money given by the fraction λ.
Naturally the conservation law $m_i + m_j = m_i' + m_j'$ must still be valid, since no money leaves or enters the system. However, the equation denoting the change in money for each agent is now given as

$$m_i' = \lambda m_i + \epsilon(1-\lambda)(m_i + m_j), \tag{4}$$

and using the conservation law, the other agents money

$$m_j' = \lambda m_j + (1-\epsilon)(1-\lambda)(m_i + m_j). \tag{5}$$

Equation 4 and 5 can furthermore be rewritten as follow

$$m_i' = m_i + \delta m \tag{6}$$

$$m_j' = m_j - \delta m \tag{7}$$

with

$$\delta m = (1-\lambda)(\epsilon m_j - (1-\epsilon)m_i). \tag{8}$$

Compared to the previous model, where no other parameters than the interaction between the agents is taken into account, the saving model results in a different wealth distribution curve, which in larger degree resembles Boltzmann-Gibbs distribution, with the very richest individuals following a power law

$$e^b x^a. \tag{9}$$

Here a is the slope of the linear fitting of the logarithmic histogram data, and b is the intercept.

It can be shown that the equilibrium distribution follows an analytical form for $(0 < \lambda < 1)$ [8]. By introducing the reduction variable

$$x = \frac{m}{\langle m \rangle} \tag{10}$$

and the parameter

$$n(\lambda) = 1 + \frac{3\lambda}{1-\lambda}, \tag{11}$$

the money distribution follows the function

$$P_n(x) = a_n n - 1 e^{-nx}. \tag{12}$$

Here the prefactor a_n is given by

$$a_n = \frac{n^n}{\Gamma(n)}, \tag{13}$$

where $\Gamma(n)$ is the Gamma function.

Nearest Neighbours and Acquaintances

Instead of picking the interacting agents at random, we instead wish to model this a bit more realistically. By adding a likelihood of transacting given by the following equation,

$$P_{ij} \propto \Big| m_i - m_j \Big|^{-\alpha}. \tag{14}$$

Like earlier, the agents i and j have corresponding amounts of money m_i and m_j, respectively. This means that the agents have a preference of transacting with other agents that are close to themselves in terms of money amount. If the difference in money between two agents is less then one, the probability of transaction will rise very quickly. At some point, the two agents may have the same amount of money, meaning $m_i = m_j$. In that case, we consider it the transaction to take place with 100 % certainty, $P_{ij} = 1$.

Former Transactions

One might argue that picking a client primarily based on your and their money capital is unrealistic, and perhaps one instead chooses to weigh whether one has done previous transactions with said agent. Taking this into consideration we add another likelihood to the equation above.

$$P_{ij} \propto \Big| m_i - m_j \Big|^{-\alpha} (c_{ij} + 1)^{\gamma}. \tag{15}$$

The variable c_{ji} is the number of previous transactions the two agents have done, and γ denotes to which degree this affects the total probability of a transaction P. The addition of 1 is there to ensure two agents can still interact, even though they have not before.

III. METHOD

We have implemented an algorithm which simulates the above transactions, depending on the variables λ, α and γ, with an initial amount m_0. At each transaction, a randomly selected pair of agents is selected, where the transaction is only accepted if the probability of them interacting with each other is larger than a random value.

Algorithm 1 Transactions

Initialize start amount m_0 for each agent and number of interactions $c = 0$
for i = 1 : number of transactions **do**
    Choose a random pair of agent (i, j)
    if $m_i == m_j$ **then**
        p = 2
    else
        p = $abs(m_i - m_j)^{-\alpha}(c_{ij} + 1)^{\gamma}$
    end if
    Generate a random number r, $r \in [0, 1]$
    if $r < p$ **then**
        $c_{ij} = c_{ij} + 1$
        Generate random reassignment ϵ, $\epsilon \in [0, 1]$
        $\delta m = (1 - \lambda)(\epsilon m_j - (1 - \epsilon)m_i)$
        $m_i' = m_i + \delta m$
        $m_j' = m_j - \delta m$
    end if
end for

In order to further understand the probability of the different outcomes after a number of transaction, we apply a Monte Carlo simulation. The Monte Carlo simulation helps to visualize all the possible outcomes, which are hard to predict due to the random variables, and give a better idea of the risk of a decision. In our case, the Monte Carlo simulation will help us to makes sure we reach the steady-state distribution. We will discuss the choice of number of transactions and Monte Carlo cycles for the different models later.

Random Numbers

As in any Monte Carlo method, a large amount of random numbers are drawn while doing the simulations. Due to this, it is important that we use a high quality random number generator. We therefore use the Mersenne Twister 19937 algorithm[1], which as the name states has a period of 2^{19937}. This should be more than enough for our simulations.

Equilibrium state

We are only interested in analyzing the steady-state distribution, where the final distribution stays approximately constant. Thus, it is critical to know how many transactions and Monte Carlo cycles are needed to reach this equilibrium state. We already know the simple model relaxes toward the equilibrium state given by the Gibbs distribution. On the other hand, when we add parameters like the saving criterion, the system does not relax towards this distribution anymore. It is then necessary to find another criterion for when the equilibrium situation has been reached.
We will look at when the variance converges for the different systems, to find the number of transactions needed. The variance is given by

$$var = \frac{\sum_{i=1}^{N} m_i^2}{N} - \frac{\left(\sum_{i=1}^{N} m_i\right)^2}{N^2} \tag{16}$$

where m_i denotes the wealth of one agent and N is the number of total agents. In addition we will look at when the relative difference in the histograms between successive Monte Carlo cycles converges towards zero to find the number of Monte Carlo cycles needed. This difference is given by

$$\delta_i = \left| \left(\frac{a_i}{i}\right)^2 - \left(\frac{a_{i-1}}{i-1}\right)^2 \right|^{1/2} \tag{17}$$

where a_i denotes the average wealth after i Monte Carlo cycles. From this we then conclude with the system to have reached the equilibrium state when we have performed enough transactions for the variance to stabilize and enough Monte Carlo cycles for the relative difference to converge towards zero.

Power law in the high end tails

As mentioned earlier we expect all but the simple model to follow a so called power law in the high end tails, as proposed by V. Pareto [3] (see equation 9), and as shown by Goswami and Sen. [4]. In doing this, we simply extract the very richest individuals, for example the very 50 or so richest agents, and use a linear fitting to extract the slope and intercept of the logarithmic data to use in equation 9. Some variation in the parameters is necessary to ensure a good linear fitting. For example the number of agents used in fitting the power law.

IV. RESULT

In the following plots, the use of time steps on the x-axis denotes the number of total interactions, both accepted and rejected. There is one time step between two possible transactions, unaffected of rejection and acceptation.

The Simple Model

The variance for increasing number of transactions and the relative difference between successive Monte Carlo cycles is shown in Figure 2 and 3.

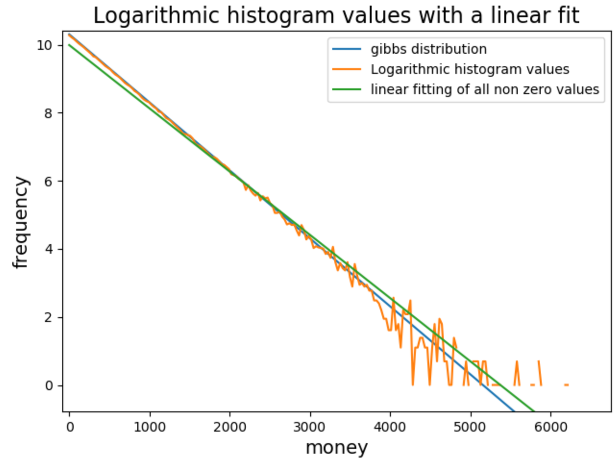

Figure 1. Histogram values for the simple model, without saving, or acquaintances. A linear fitting is shown, which only fits nonzero values. The increase in noise is due to the fact that the plot is logarithmic. In order to show that the data follows Gibbs distribution, we have also included this given by equation 3

The Saving Model

To find the point where the model reaches equilibrium state, we will look at the variance and relative difference in the histogram for successive Monte Carlo cycles. Figure 2 and 3 include data for both the simple model and the saving model.

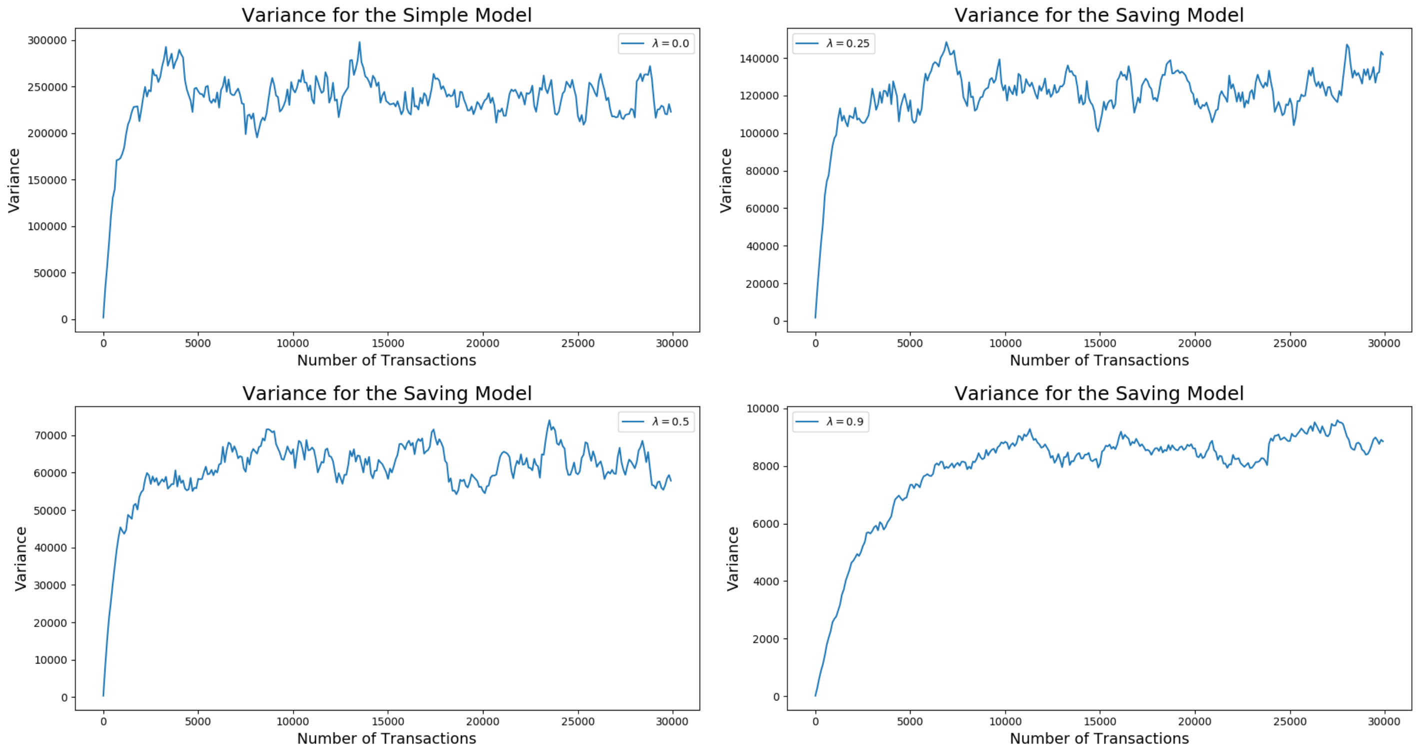

Figure 2. Variance for the simple model and the saving model with 500 agents with a start wealth of 500. The top left plot shows the variance for the simple model with $\lambda = 0$, while the others show the variance for the saving model with saving fractions $\lambda = 0.25, \lambda = 0.5$ and $\lambda = 0.9$. The variance converges faster for smaller saving fraction λ.

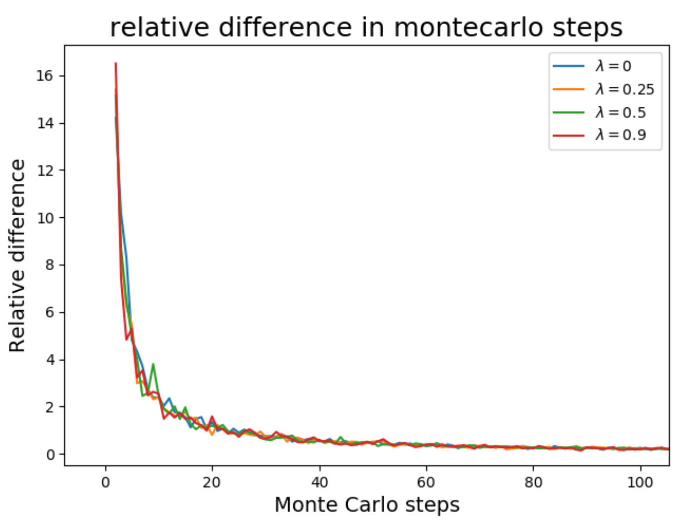

Figure 3. The relative difference in the histograms between successive Monte Carlo cycles for the different values of λ. It clearly converges towards zero after 100 Monte Carlo cycles.

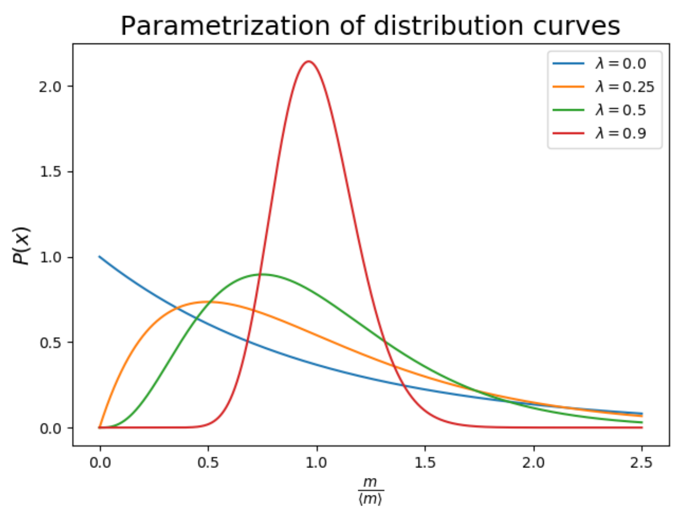

Figure 4. Parameterization of the distribution curves. When the saving fraction λ increases, the distribution goes from following a Gibbs distribution to get closer to a Gaussian distribution.

We have then parameterized the distribution curves using equation 12 and plotted the corresponding distribution curves.

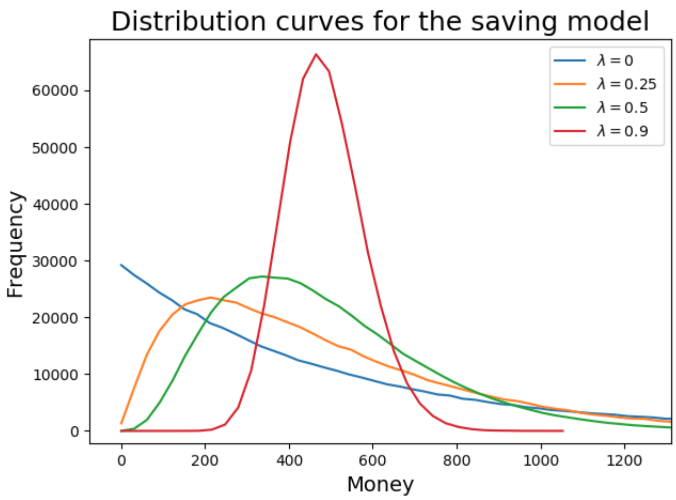

Figure 5. Plot of the distribution curves for the simple and saving model with 10^7 transactions and 10^3 Monte Carlo cycles.

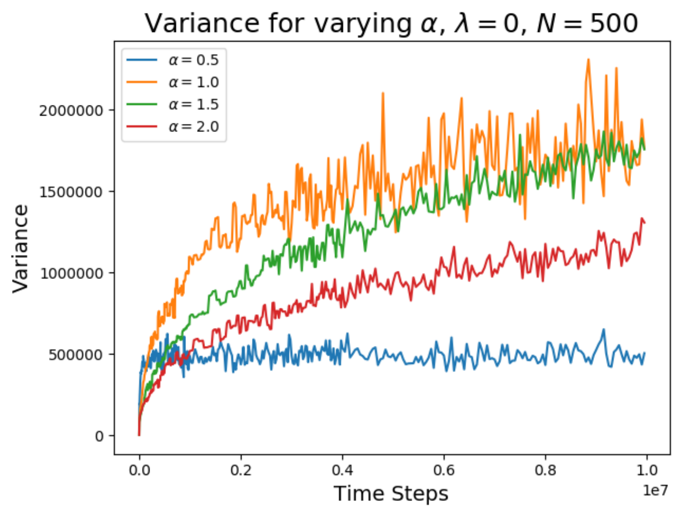

Figure 7. The variance as a function of time steps for varying α and fixed $\lambda = 0$ and $\gamma = 0$. After 10^7 transactions the variance has stabilized for $\alpha = [0.5, 1]$, but not for $\alpha = [1.5, 2]$.

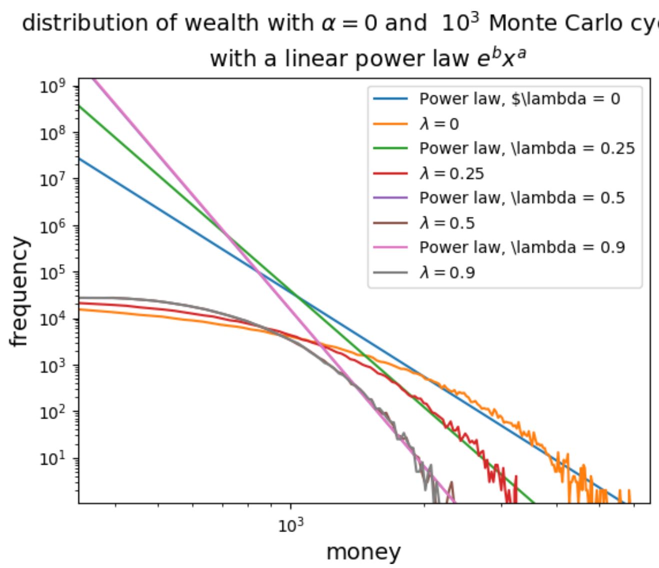

Figure 6. The high end tail of the money distribution for various λ and $\alpha = 0$. an exponential fitting has been done with the last 50 datapoints, to show that these follow a power law, as proposed by V. Pareto [3]. Note: The data and power law of $\lambda = 0.5$ is hard to spot, since it is hidden underneath the plot of $\lambda = 0.9$.

Nearest Neighbours and Acquaintances

We have already seen that the variance converges slower for a larger saving fraction, hence we find it the most interesting to see how many transactions are needed for the variance to stabilize for varying α when the saving fraction λ equals 0.9. We also look at the difference between a system with 500 and with 1000 agents.

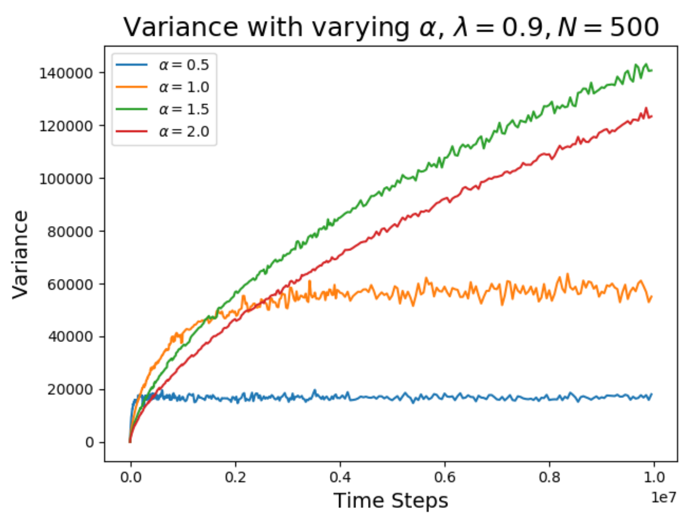

Figure 8. The variance as a function of time steps for varying α and fixed $\lambda = 0$ and $\gamma = 0$.After 10^7 transactions the variance has stabilized for $\alpha = [0.5, 1]$, but not for $\alpha = [1.5, 2]$.

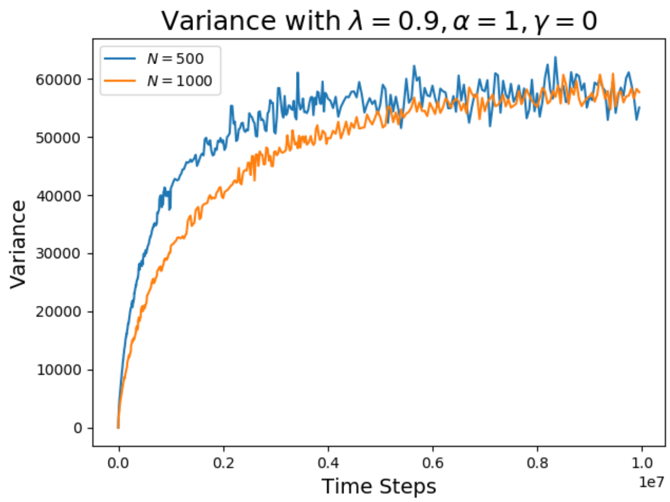

Figure 9. The variance as a function of time steps for varying N with fixed $\lambda = 0.9, \alpha = 1$ and $\gamma = 0$. The variance stabilizes slower for $N = 1000$ than $N = 500$.

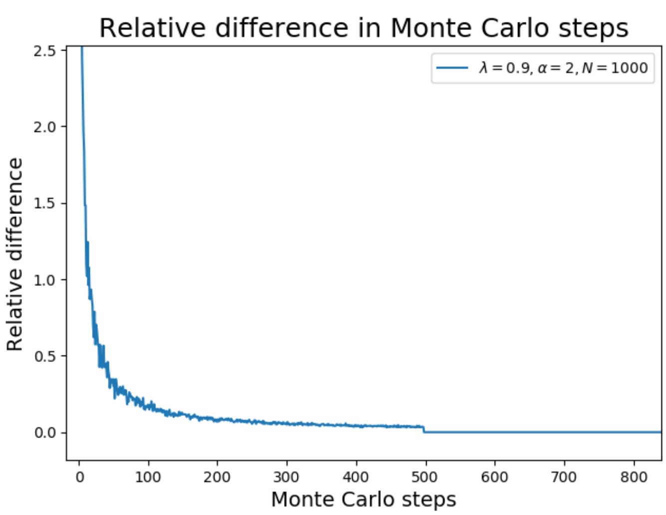

Figure 10. The relative difference in the histograms between successive Monte Carlo cycles for the different values of λ after 10^7 transactions. It clearly converges towards zero after 500 Monte Carlo cycles.

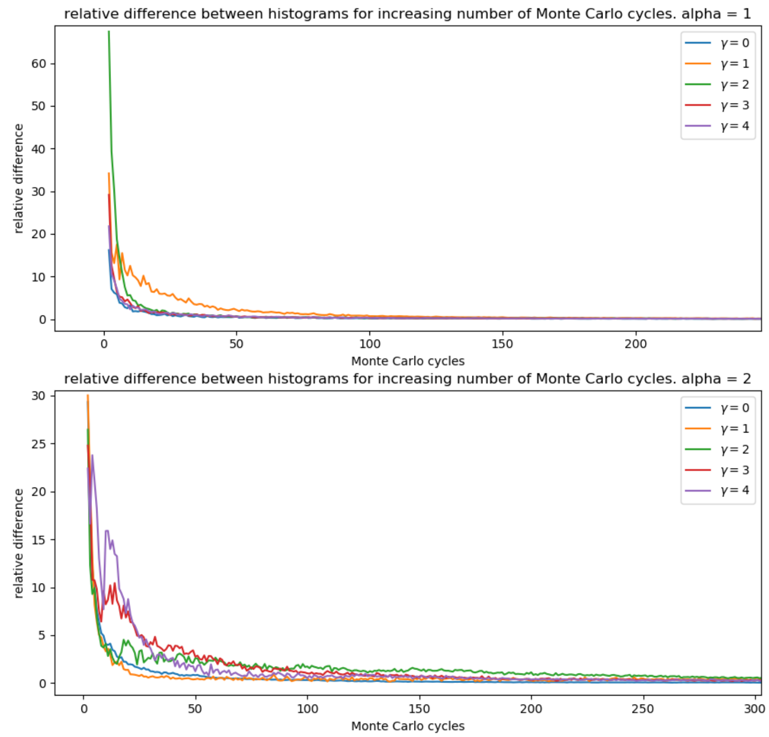

Figure 12. The relative difference in the histogram for increasing amounts of Monte Carlo cycles. Here, 1E7 transaction were done in each Monte Carlo cycle, with 1000 agents and no saving, I.E. $\lambda = 0$

Former Transactions

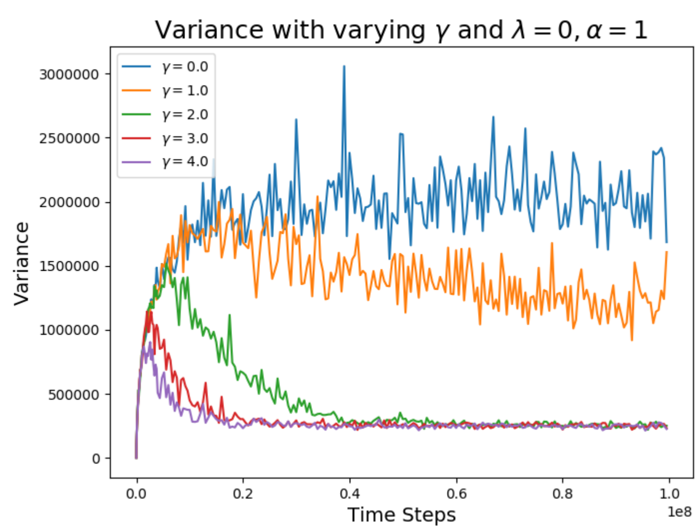

Figure 11. The variance as a function of time steps for varying γ and fixed $\lambda = 0$ and $\alpha = 1$ for $N = 1000$. The variance stabilizes faster for higher value of γ.

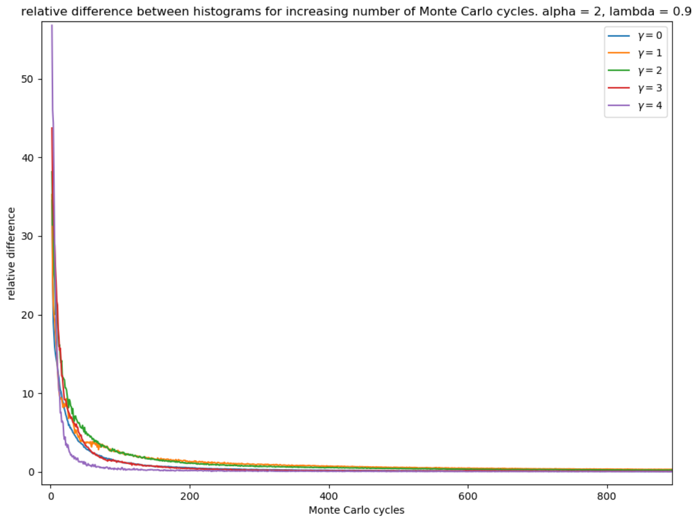

Figure 13. The relative difference in the histogram for increasing amounts of Monte Carlo cycles. Here the values $\lambda = 0.9$ and $\alpha = 2$ is used, since these variables caused the variance to converge towards a stable solution the slowest. Since we see they reach a steady state within 1000 Monte Carlo cycles we can still conclude that the method reaches a steady solution.

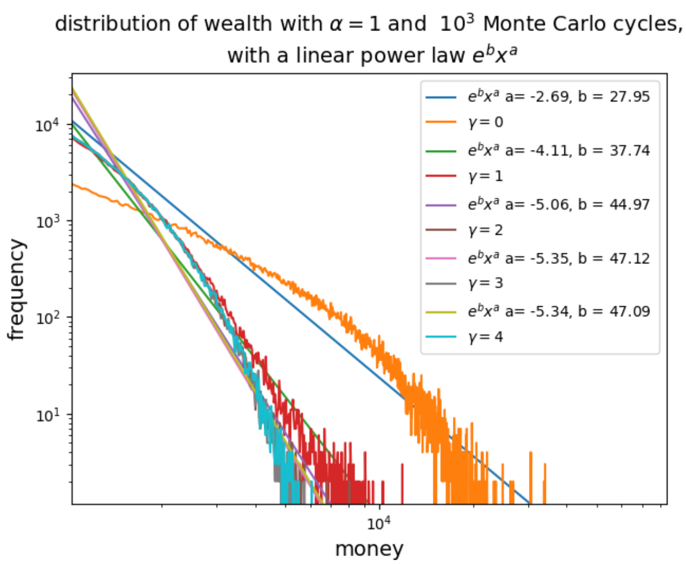

Figure 14. The various high end tails for $\alpha = 1$, with a power law shown with corresponding values for the slope and intercept a and b

V. DISCUSSION

The Simple Model

From Figure 2, $\lambda = 0$, we see that the variance converges fast and is more or less stable after 10^4 transactions. Since we perform the analysis on 500 agents, this means the agents on average make $2 \cdot 20 = 40$ transactions at this point, so they all interact at some point in the simulation. To guarantee the distribution reaches equilibrium state, we increase the number of transaction to 10^5.

Figure 3 shows the relative difference in the histograms for increasing numbers of Monte Carlo cycles. As the number of cycles increases the relative difference can be seen to drastically decrease, and converge towards zero. This indicates that the method converges towards a solution as desired. After 100 cycles the relative change is fairly low, so in order to guarantee that we always reach an accurate final solution we use 1000 cycles when producing our histograms.

Based on the above arguments, it is efficient enough to use 10^5 transactions and 10^3 Monte Carlo cycles to find the steady-state distribution for the simple model, and compare it to the Gibbs distribution.

In figure 1 the histogram is plotted logarithmically, with a linear fitting of the logarithmic values, which only includes the nonzero data points. This is due to the fact that any negative values are not handled, which will affect the accuracy of the linear fitting. The linear fit clearly shows that the logarithmic values follow a exponential distribution given by the Gibbs distribution $\beta e^{-\beta m}$, which is also included in the plot. The increase in noise in the plot is due to the fact that the data is plotted logarithmically, but also because the high end tails don't follow Gibbs distribution [[5], [6]]. This results in the deviation in the data to be much more clear for lower y values.

The Saving Model

From Figure 2 it is clear that the variance converges slower for a larger saving fraction. For $\lambda = 0.9$ the variance stabilizes closer to 20^3, twice as many as for the simple model.We can see that the variance decreases for larger saving fraction. This can be explained due to the smaller amount of money available in the interactions. When the agents save 90% of their money before making a transaction, there will be less money in motion, and the difference in wealth will decrease. This can also be seen as the distribution gets closer to a normal distribution centered around m_0 (discussed below). The relative difference in the histograms for increasing number of Monte Carlo cycles is the same as for the simple model, hence it is still efficient enough to use 10^3 Monte Carlo cycles for our simulations.

Since we have not normalized our data, the axis for Figure 4 and 5 are scaled differently. It is still clear that our distribution curves has the same shape as the parameterized curves. $\lambda = 0$ is equivalent to the simple method, which follows Gibbs distribution, while $\lambda = 1$ is equivalent to our initial state. This means the distributions should get closer to a distribution where everyone has the initial amount of money, m_0, as λ increases. We observe in Figure 5 that we get an almost normal distribution centered around 500, m_0 in our case, for $\lambda = 0.9$, which support the expectations.

In figure 6 we find that the last few data points in the histogram indeed follow a power law given by $e^b x^a$, where a is the slope of the linear fitting, and b is the intercept. This is in agreement with the findings in V. Paretos article [3], as well as the findings in Goswami and Sen [4].

Nearest Neighbours and Acquaintances

As we have discussed previously, the variance converges slower for higher saving fraction. We then decided to look at the number of transactions needed for the variance to stabilize without saving (Figure 7) and with a high saving fraction (Figure 8) while varying α. It is clear for both that a higher value of α requires a higher number of transactions for the variance to converge.

For $\alpha = [0.5, 1]$, the equilibrium state is reached rather fast compared to $\alpha = [1.5, 2]$. This can be explained by looking at equation 14. The variable α denotes the probability of a transaction going through, based on how close two agents are to each other in financial capital. In layman's terms the alpha denotes how picky the agents are to deal with another agent who is not on their level money wise. If we for example use a very large α, the very richest people would only have a very limited number of agents to trade with, meaning they wouldn't be able to gain money very quickly. We see this reflected in the plots, where the variance for larger values of α increase slower, as the richest agents have fewer agents to transact with.

In figure 9 the variance for two different simulations with different numbers of agents are shown. The larger number of agents result in a slower increase of variance, perhaps contrary to the expected behaviour. The reason for this can be explained by the fact that the number of time steps is actually the total number of possible transactions. I.E. accepted and rejected transactions. When the number of agents is larger, the number of possible transactions in-between a certain agent being picked increases. This is not really true, since the agents which interact are picked at random, but on average over a large number of time steps it will become true. This means that the number of time steps before a given agent has the chance to get more money is larger, and hence, it takes longer for the richer agents to get richer, and increasing the variance.
Even though the system does not reach the equilibrium state after 10^7 transactions for $\lambda = 0.9, \alpha = 2, N = 1000$, the relative difference in Monte Carlo steps still converges towards zero according to figure 10. This means the system is stable although the equilibrium state has not been reached yet. The relative difference converges slower compared to the saving model, but we can still continue to use 10^3 Monte Carlo cycles. Hence, we find it good enough to use 10^7 transactions and 10^3 Monte Carlo cycles.

WHAT HAPPENS IF $\alpha >> 1$

Former Transactions

By adding the parameter γ, the variance changes a lot. It reaches a top before it decreases again and stabilizes. We can see in Figure 11 that system reaches equilibrium state faster for higher value of γ, and $\gamma = [2, 3, 4]$ all stabilize at the same level. The latter three values of γ are all stabilized after $5 \cdot 10^7$ transactions. With this in mind, the number of transactions should increase to 10^8 to make sure we get the steady-state distribution.

In figure 12 the relative difference in the histogram for increasing amounts of Monte Carlo cycles is shown. For $\alpha = 1$, the method converges very quickly, whilst for $\alpha = 2$ the method uses a little longer before settling essentially at zero. Whilst these plots would indicate that a steady state is reach after a few hundred Monte Carlo cycles, we still used 1000 in all our data, in order to ensure a steady state has been reached.

As with the nearest neighbours and acquaintances we look at the realtive difference in succsesive histograms for an increasing amount of Monte Carlo cycles. This is shown in figure 13. Despite the fact that $\lambda = 0.9$ and $= 2$ had the most trouble converging when we only looked at the variance (fig 8) we can see from the relative difference that the method clearly is at a stable solution despite this. All of the values of γ are well within desired range of zero, and all converge fairly rapidly.
Figure 14 shows the high end tails, as mentioned earlier, with corresponding power law fittings. As earlier, only the non zero values are used in fitting the power law. As expected, the distribution of money largely follows a Gibbs distribution in the main body, I.E. the large majority of money distributions not at either extreme of the wealth spectrum. The tails, as expected can be seen to fit nicely with the linear power law plotted. These findings are in agreement with those of many previous papers, such as Goswami and Sen [4] and N. Ding and Y. Wang [5].

VI. CONCLUSION

From the various observations done with the relative difference in Monte Carlo cycles, and the converging variance we can conclude that our method does indeed produce steady states, which bodes well for the algorithm in general. When introducing new parameters to the simulation, it is necessary to increase the number of transaction to make sure the equilibrium state is reached. On the other hand, the number of Monte Carlo cycles can stay about the same for the different models and still get a stable system.

The algorithms behaved as expected, and our results were mostly in agreement with the findings of previous papers, which have been referenced thorough the paper. The simple model did indeed produce a wealth distribution which followed the Gibbs distribution, however they did not have a power tail. The saving, Nearest neighbours and acquaintances and former transactions methods did as expected produce a distribution which featured a power law in the higher en tails, and a Gibbs distribution in the majority of the data.

In future explorations one should look closer at the Pareto exponent has been an important value for identifying and understanding the power tail which is prominent in the distribution in many of the models. It would also be of interest to include more realistic parameters, such as taxation, to simulate a more socialist system, such as the Scandinavian model.

APPENDIX

[1] Makoto Matsumoto and Takuji Nishimura. 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. 8, 1 (January 1998), 3-30. DOI: https://doi.org/10.1145/272991.272995

[2] Jensen, M. H., Lecture in FYS3150 29/8-19.

[3] V.Parteo, Cours d'economie politique, Lausannem 1897

[4] Goswami and Sen. Agent based models for wealth distribution with preference in interaction, physica A volume 415 december 2014. https://www.sciencedirect.com/science/article/pii/S0378437114006967

[5] A.C. Silva, V.M. Yakovenko, Eur. Phys. J. B 20 (2001) 585. https://iopscience.iop.org/article/10.1209/epl/i2004-10330-3

[6] N. Ding, Y. Wang, Chin. Phys. Lett. 24 (2007) 2434. https://www.scopus.com/record/display.uri?eid=2-s2.0-34547569396&origin=inward

[7] A. Chatterjee, B.K. Chakrabarti, Eur. Phys. J. B 60 (2007) 135. https://link.springer.com/article/10.1140%2Fepjb%2Fe2007-00343-8

[8] M. Patriarca. A. Chakraborti, K. Kaski, Physica A 340, 334 (2004) https://www.sciencedirect.com/science/article/pii/S037843710400432