

UiO : **Department of Physics**
University of Oslo

Monte Carlo Modeling of Transactions

**Erik Skaar
Mikael Kiste
Thomas Storaas**



Contents

1	Introduction	2
2	Physical theory	3
2.1	Basics	3
2.2	Savings	4
2.3	Willingness to trade	4
2.3.1	Normalization	4
2.4	Familiarity and trust	5
2.4.1	Normalization	6
3	Computational theory	7
4	Implementation	8
5	Result & Discussion	9
5.1	5a & 5b	9
5.2	5c	10
5.3	5d	11
5.4	5e	14
6	Conclusion	16
7	References	17
8	Appendix	17

Abstract

1 Introduction

Predictions of how the financial market works has, in the modern age, become a large field of study[5]. With the knowledge of how stock exchanges work can a mathematical model be built up to simulate these exchanges in the hope that one can predict prices and trends in the future. These predictions are in turn used as a basis for investments and such. The accuracy and speed of these models are therefore of the utmost importance. When these models predict the wrong outcome things can go very wrong as it did in 2009[2].

In this project a simulation is done on a simple closed economy model. The economy model consists of a group of financial agents with the ability to transfer money between themselves, using the Monte Carlo method. The simulation will be compared to the well established Pareto's law [3], and the analysis follows two papers. Patriarca with collaborators made one of the articles.[4] The second one was written by Goswami and Sen. [1]

In the first simulation all financial agents are made equal and there is a 100% probability of a transaction taking place when a trade is suggested. This is the simplest case and the model is then compared with a logarithmic plot to verify the trend. Then complexity is added by introducing saving. The model is further refined by adding a willingness to trade with agents with a comparative wealth to themselves. Lastly a favorable bias towards agents in which a trade has happened before is added. The whole job will be parallelized to speed up the simulations.

legg til noe mer for å dekke en side?

2 Physical theory

2.1 Basics

From empirical studies Pareto has been shown that the higher end of a distribution of money follows the distribution trend:

$$\omega_m \propto m^{-1-\alpha} \quad (1)$$

Here $\alpha \in [1, 2]$. This trend will be analytically studied with a system of micro dynamical relations between a group of financial agents, and the resulting money distribution between them.

The system consists of N financial agents. They transfer money between them in pairs of (i, j) and they all start with a fixed amount of money m_0 . The amount of money is arbitrary. So in this project the amount of money is given in amount of "wealth". Here one wealth is m_0 and the symbol used is \$. A result from this is that $\langle m \rangle = 1\$$

At a time step a pair of agents, (i, j) , are chosen at random and let them transact money from j to i . The pair is chosen at random so for a specific pair m_i, m_j both (m_i, m_j) and (m_j, m_i) are possible. Money is conserved during the transaction which means that the money Δm transferred from j to i has to be equal. This means that the total expression for the transaction can be written as

$$m_i + m_j = m'_i + m'_j \quad (2)$$

This transfer is done by a Randomized Number Generator and in this model none of the agents are left with a debt. To remove the debt from the picture, only a percentage of the money is transferred instead of a fixed amount.

$$m'_j = \epsilon(m_i + m_j) \quad \epsilon \in [0, 1] \quad (3)$$

And similar for m'_i .

$$m'_i = (1 - \epsilon)(m_i + m_j) \quad \epsilon \in [0, 1]$$

Due to the limitations of ϵ no agent gets debt, but it is likely that one agent is left with zero $m \geq 0$. This system now consists of a conserved amount of money with a transfer criteria. It can be shown that this system, given enough time, will relax towards a Gibbs distribution.

$$\omega_m = \beta e^{-\beta m} \quad (4)$$

With

$$\beta = \frac{1}{\langle m \rangle} = \left[\sum_i \frac{m_i}{N} \right]^{-1} \quad (5)$$

This model uses $N = 500$. The 500 agents has to do enough transaction to achieve the above outlined distribution. The article suggests that 10^7 transactions is enough. [4] To find the final equilibrium distribution ω_m 10^4 runs were done and ω_m was taken as the average of all the recorded ω_m . The distribution of agents is exponential, so plotting the logarithm of the trend should yield a straight line.

$$\begin{aligned} \ln(\omega_m) &= \ln(\beta e^{-\beta m}) \\ \ln(\omega_m) &= \ln(\beta) - \beta m \end{aligned}$$

2.2 Savings

To make the model more realistic complexity is added though a savings term. Now every agent will save a fraction of their wealth δ before the transaction takes place. The addition of this term will change the distribution of agents so it no longer makes a gibbs distribution.

$$m'_i = \lambda m_i + \epsilon(1 - \lambda)(m_i + m_j) \quad (6)$$

and similar for m'_j .

$$m'_j = \lambda m_j + (1 - \epsilon)(1 - \lambda)(m_i + m_j) \quad (7)$$

The two terms in the expressions can be simplified by gathering both the savings term and the trading term in a variable $\delta m = (1 - \lambda)(\epsilon m_j - (1 - \epsilon)m_i)$

$$m'_i = \lambda m_i + \delta m$$

$$m'_j = \lambda m_j + \delta m$$

2.3 Willingness to trade

The model has up until now only varied how much is saved and not touched on if they want to trade. From observations it is known that the willingness to trade impacts if a trade takes place or not. Further refinement of the model includes two terms to include in the probability of trading. For this the model follows the work of [1]. The model now takes into account that there is a probability

$$p_{ij} \propto |m_i - m_j|^{-\alpha} \quad (8)$$

for an interaction to take place between two agents with respectively m_i and m_j wealth. The variable α is larger than zero. If α is zero one gets the same model as for ???. Several runs will be done with varying α to analyze the effect of the probability term on the expression.

2.3.1 Normalization

The expression for the willingness to trade ??? is not normalized. If one chooses a value less than one the expression will blow up and the probability becomes larger than one.

To prevent unrealistic probabilities above one a normalization criteria is introduced. The suggested normalization criteria on piazza is to normalize it with the expectation value of the agents $\langle m \rangle$.¹ From ??? the expectation value $\langle m \rangle = 1\$$. So dividing by this will not change this expression. The other change is to add a factor of two to the probability, $2|m_i - m_j|^\alpha / \langle m \rangle$ because the normalization gives an overall decrease in all the probabilities. This is not included because of the way the expectation value is defined here gives no change in the probability. The last piece in the normalization is to remove the probabilities higher than one, although it is not important for this part it will be important in the next section. All probabilities higher than one is synthetically set to one. The physical representation of

¹Piazza: Morten's suggestion

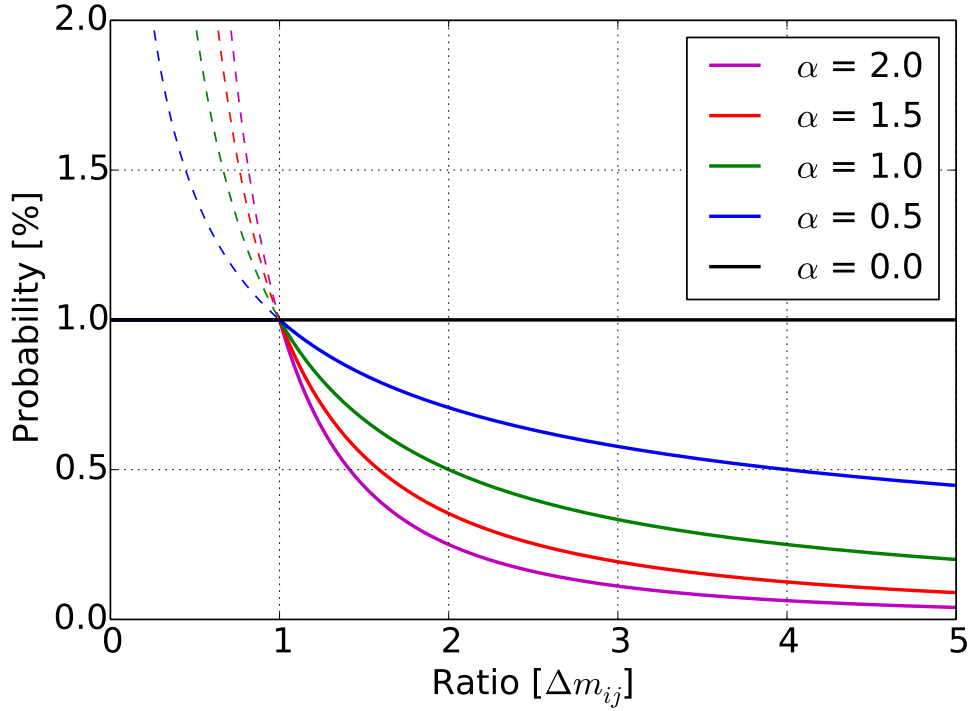


Figure 1: Figure shows the probability as a function of the distance in wealth between the two agents. The continuous line is the probability chosen in this project while the dashed line shows the trends without the criteria below one.

this is that if two agents has a difference in wealth that is less than one, a trade is guaranteed.

$$p_{ij} = \begin{cases} |m_i - m_j|^{-\alpha} & |m_i - m_j| > 1 \\ 1 & |m_i - m_j| < 1 \end{cases} \quad (9)$$

2.4 Familiarity and trust

From observations it is known that it is more likely that two agents that know and trust each other trades. In this model this is introduced to the system by adding to the probability a increased probability between agents that has traded earlier.

$$p_{ij} \propto |m_i - m_j|^{-\alpha} (c_{ij} + 1)^\gamma \quad (10)$$

c_{ij} represents the number of previous transactions between the two agents. If $c_{ij} = 0$ then the $+1$ factor makes sure that the trust term does not influence the probability. Several runs will be done with varying γ to analyze the effect of the modified probability term on the expression.

2.4.1 Normalization

As for the term in 2.3.1, the expression above has to be normalized. The normalization criteria suggested in piazza is to divide by the highest amount of trade two agents has traded.²

$$p_{ij} = |m_i - m_j|^{-\alpha} \left(\frac{c_{ij} + 1}{c_{\max} + 1} \right)^\gamma \quad (11)$$

By doing this one would make sure that no value supersedes one.

An alternative normalization criteria is to rather than taking the total of all the transactions one could normalize by the average of the two agents maximum value.

$$p_{ij} = |m_i - m_j|^{-\alpha} \left(\frac{c_{ij} + 1}{\langle c_{ij, \max} \rangle + 1} \right)^\gamma \quad (12)$$

The physical representation of this is that when an agent makes a trade it is with respect to what he has previously done, not what the maximum of all the agents. This will increase the overall probability of a transaction taking place.

disk

$$p_{ij} = \begin{cases} |m_i - m_j|^{-\alpha} \left(\frac{c_{ij} + 1}{\langle c_{ij, \max} \rangle + 1} \right)^\gamma & |m_i - m_j| > 1 \\ \left(\frac{c_{ij} + 1}{\langle c_{ij, \max} \rangle + 1} \right)^\gamma & |m_i - m_j| < 1 \end{cases} \quad (13)$$

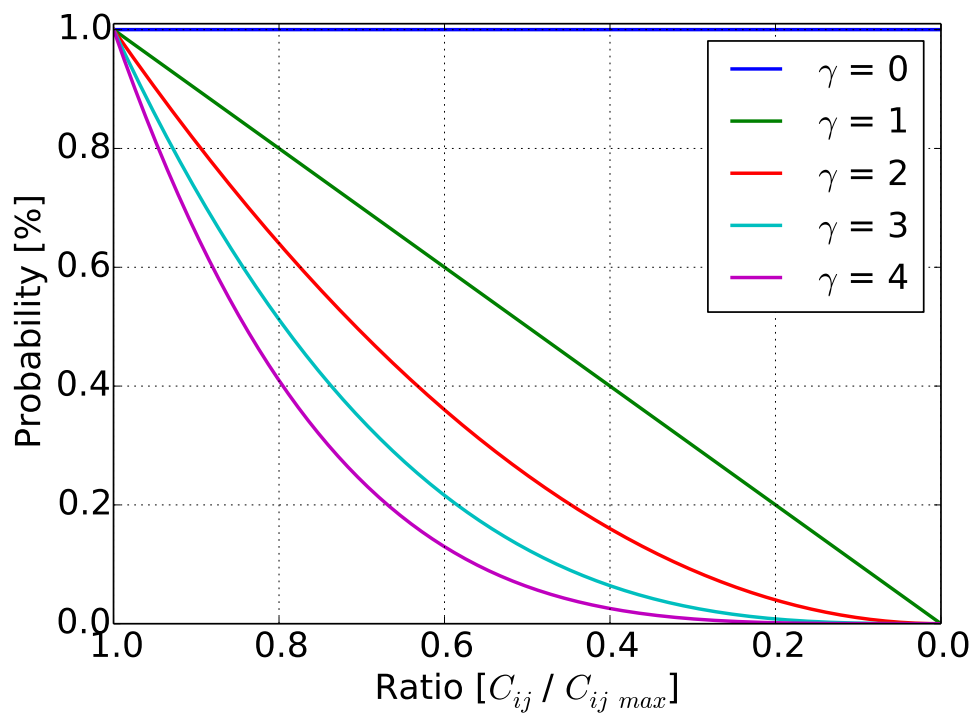


Figure 2: Figure shows the probability as a function of their trade ratio.

²Piazza: Morten's suggestion

3 Computational theory

This project consists of four different cases with a different degree of complexity. For the basis case and the savings case it is enough to use the Monte carlo method. The Monte Carlo method is a method that utilizes the fact that a large number of experiments converges towards the expectation value. When the probability of doing a transaction is added the system utilizes the Metropolis Algorithm is implemented. The system has to do a random choice of acceptance or denial of the trade. This random choice is done with the Random Number Generator. A random number i created between $(0 - 1)$ If the number is less than the probability term, accept the new state. If the number is greater than the energy change then discard the change. The procedure is outlined in four steps.

- Chose two agents at random.
- Find the probability of acceptance for this new case
- A RNG is used to chose a number between $(0 - 1)$. Now if the probability term is less then the RNG number, reject the transaction and discard the trade. If not, accept the trade.
- Choose a new random number for how much is traded between the two(ϵ).
- Update the system with the transfered money.

4 Implementation

The algorithm was implemented as discussed in section ?? in the program called main.cpp. main.cpp has some functions for different versions of the stock market. The difference between the versions are how likely it is that two random agents will trade. For more insight see section ??.³

Many different simulations is needed for this report. This is why a parallelized version has been made. MPI is used since it is easy to implement. Most of the code in the parallelized version is the same as for the non-parallelized. The parallelized version should then be X times faster then the normal version with X cores. Each thread and runs the simulations for some given values for α , λ and γ . The values is determined by the rank of the thread and will therefor be unique to other thread's values.⁴

Table 1: The agents were given 10^7 opportunities to do a transaction and the system was simulated 160 times. The test ran on a macbook pro 15. It has a quad core CPU. Expected difference is 4. The difference was higher, then expected. This is due to the Hyper-Threading technology in this CPU.

Version	Normal	MPI	Expected difference	Actual difference
A	35.147 s	7.361 s	4.000	4.774
D	98.112 s	17.202 s	4.000	5.764
E	226.522 s	45.608 s	4.000	4.966

Table 2: For scaling we expect a time increased proportional to the size increase. The table shows how we expect the time to develop and how it actually it develops. There is a minor difference from expected and calculated and that comes from the fact that the program does more then just the algorithm and the fact that the algorithm has not been perfectly implemented.

Version	# Simulations	Expected time	Actual time	T_i / T_{10}
A	10	x	2.252 s	1.000
A	160	16x	35.147 s	15.60
D	10	x	6.229 s	1.000
D	160	16x	98.112 s	15.75
E	10	x	12.876 s	1.000
E	160	16x	226.522 s	17.59

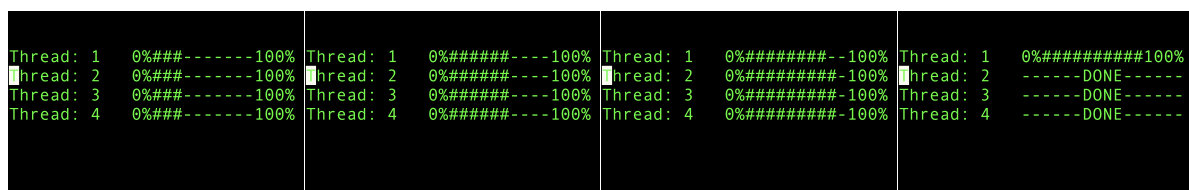


Figure 3: A progress bar was made for the program. It shows the progress for each individual thread. The progress bar works for 1 to 4 threads. More then 4 threads might look funny. By the way, we are A students in the newly announced PRO101 subject.

³All of the programs discussed in this section can be found at [github](#)

⁴[Intel Hyper-Threading Technology](#)

⁵ [PRO101: Procrastination for students](#)

5 Result & Discussion

5.1 5a & 5b

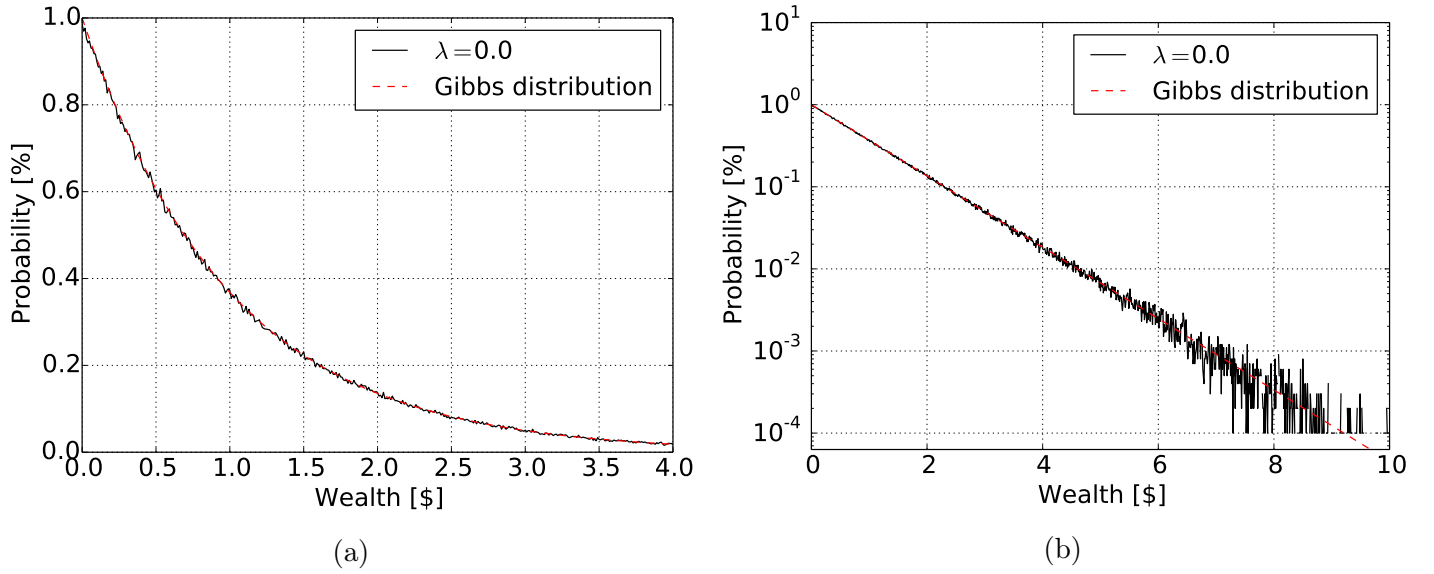


Figure 4: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

5.2 5c

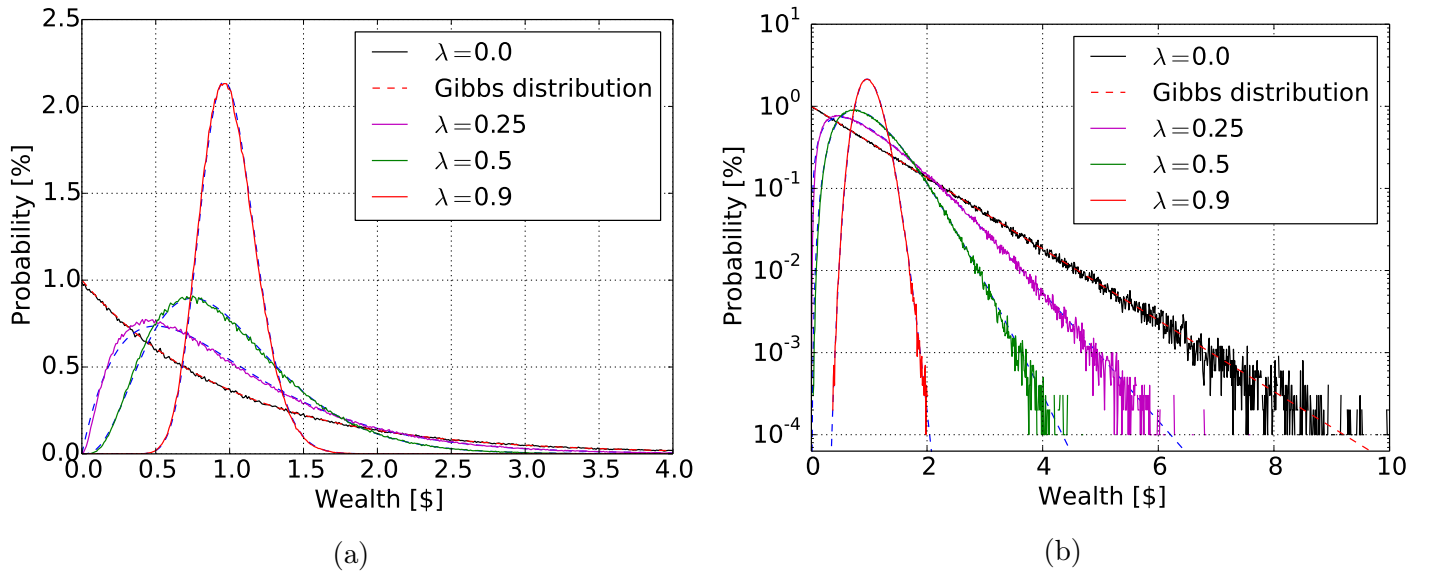


Figure 5: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

5.3 5d

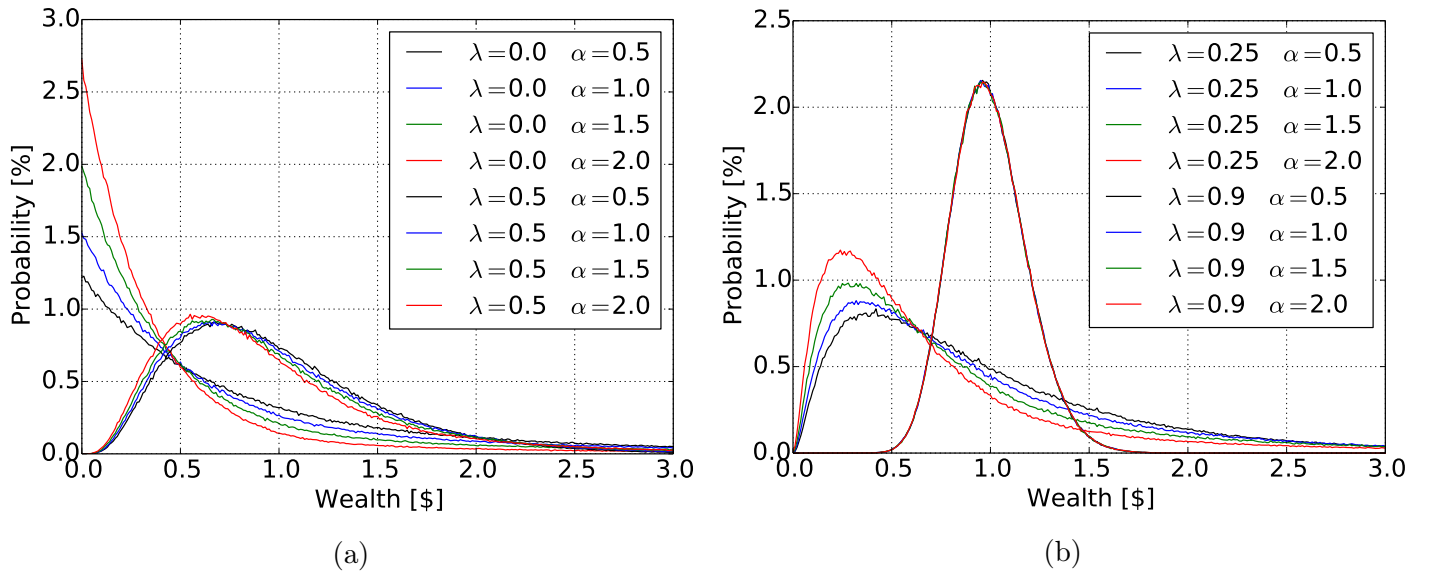


Figure 6: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

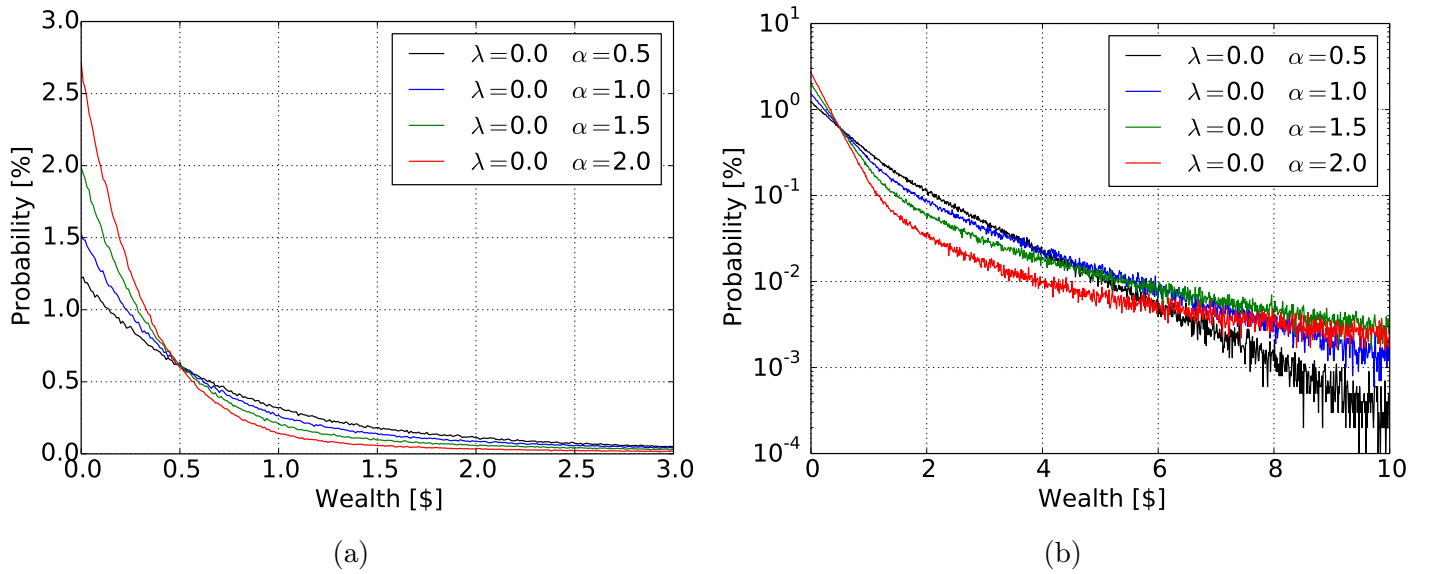


Figure 7: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

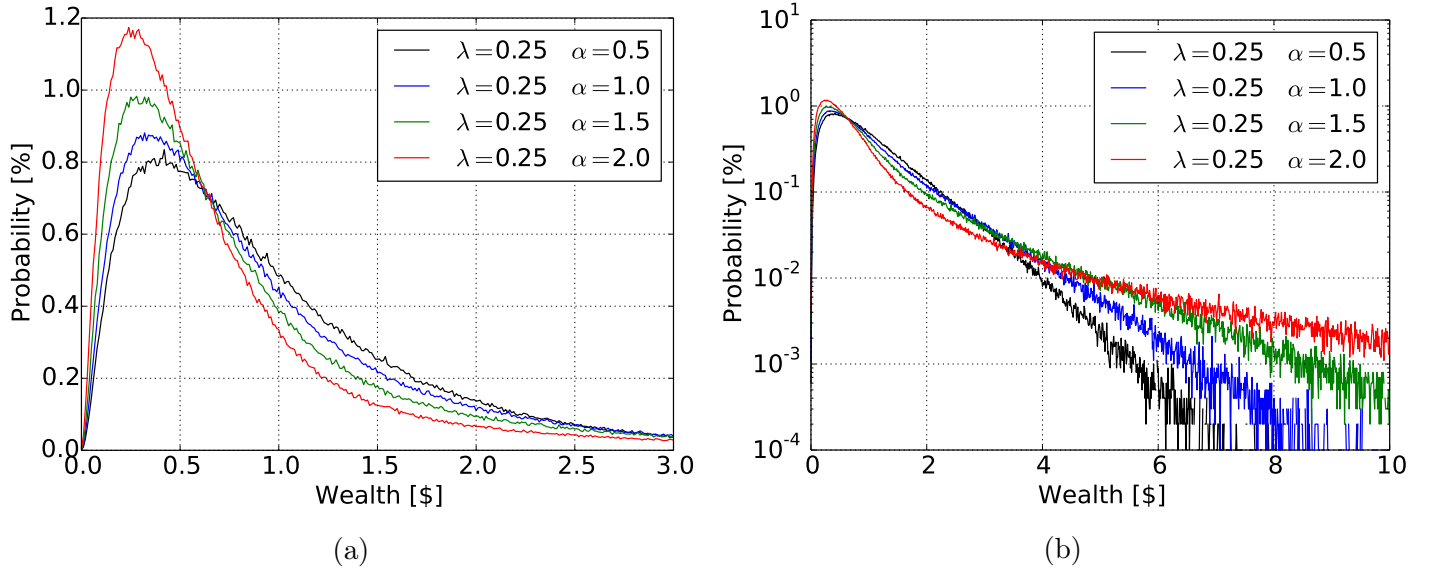


Figure 8: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

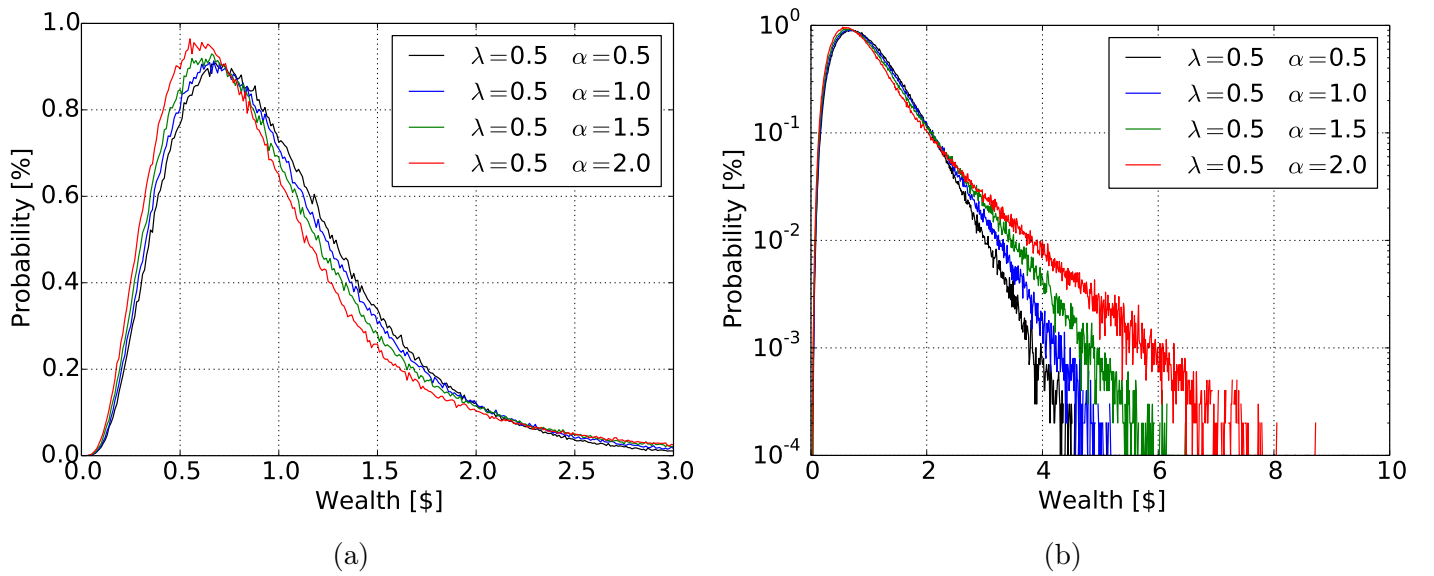


Figure 9: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

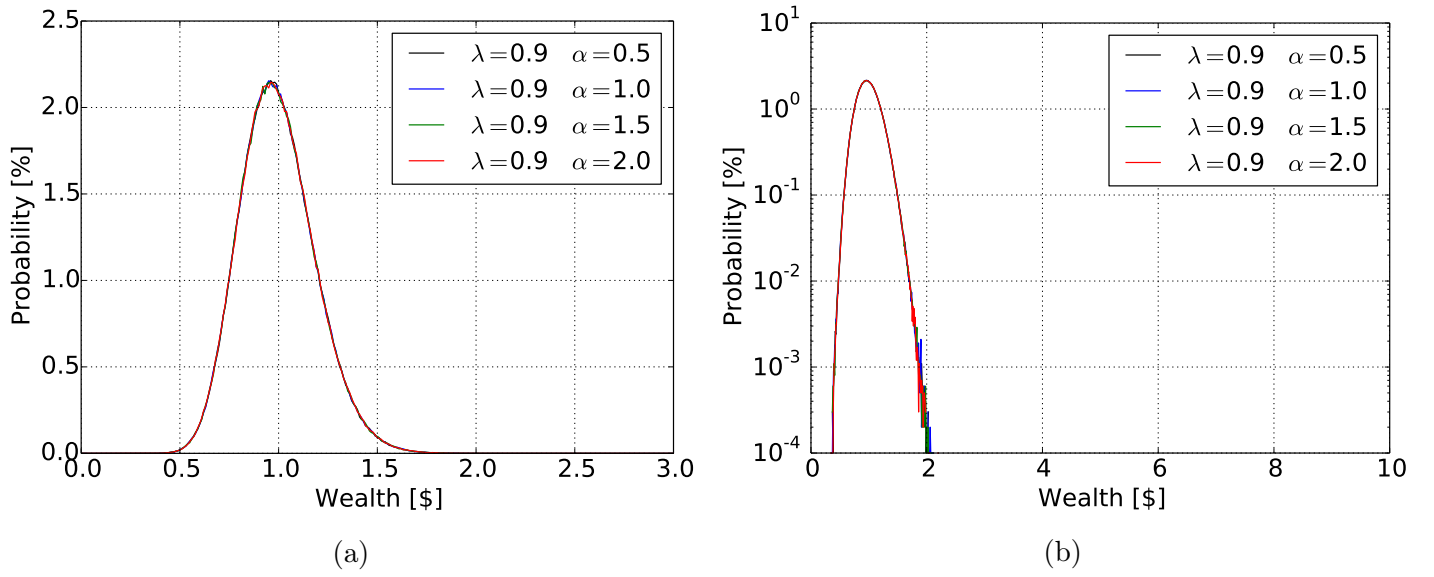
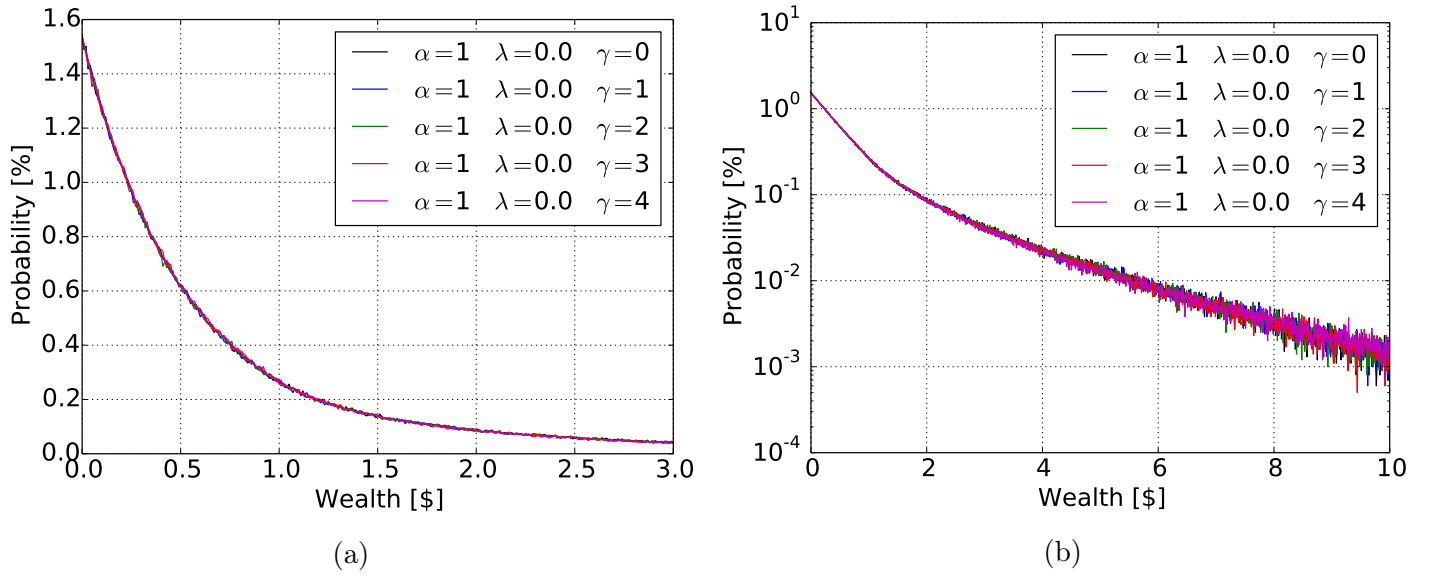
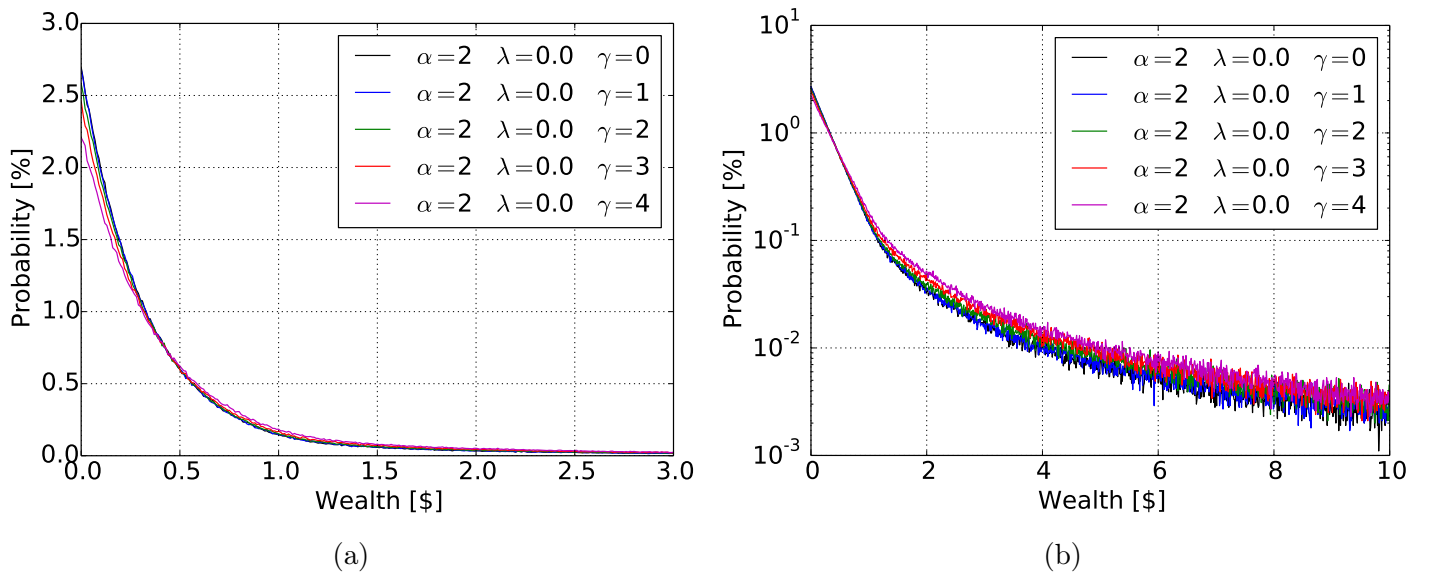
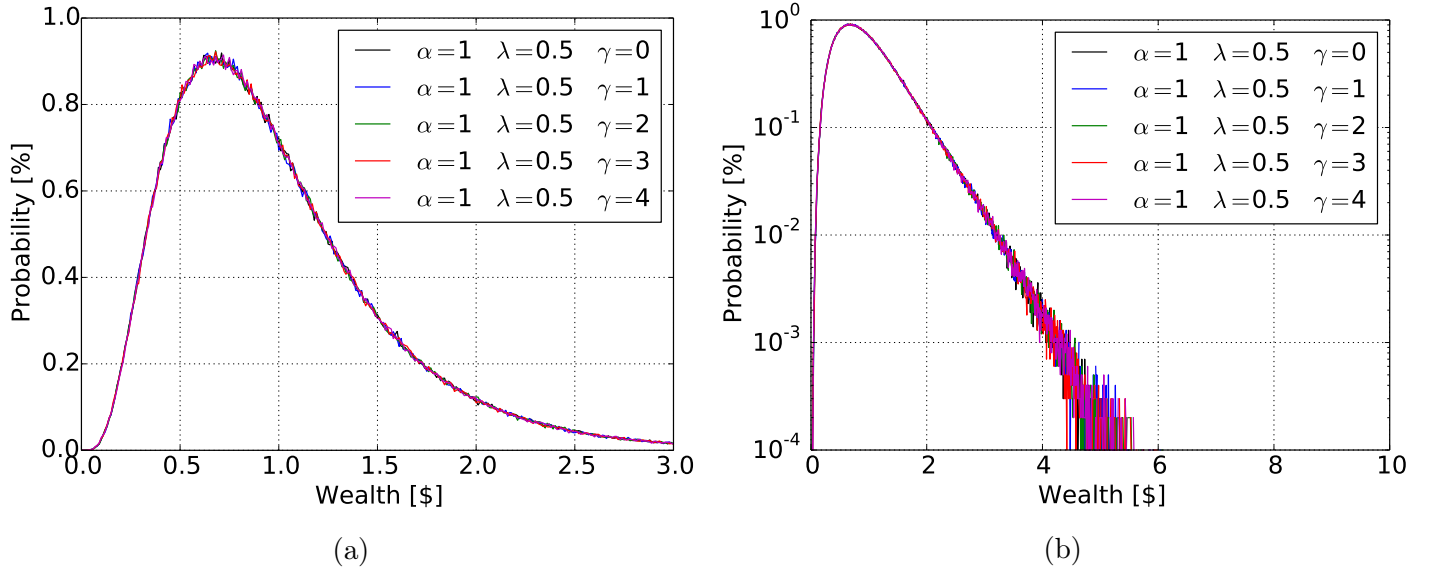
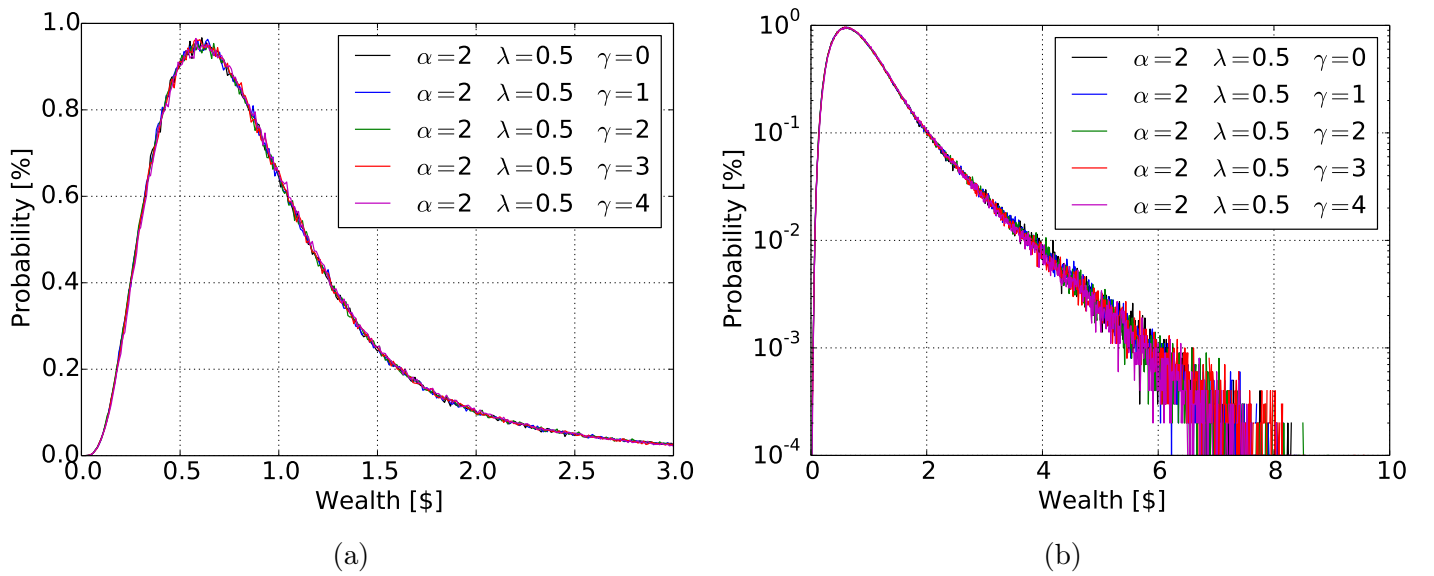
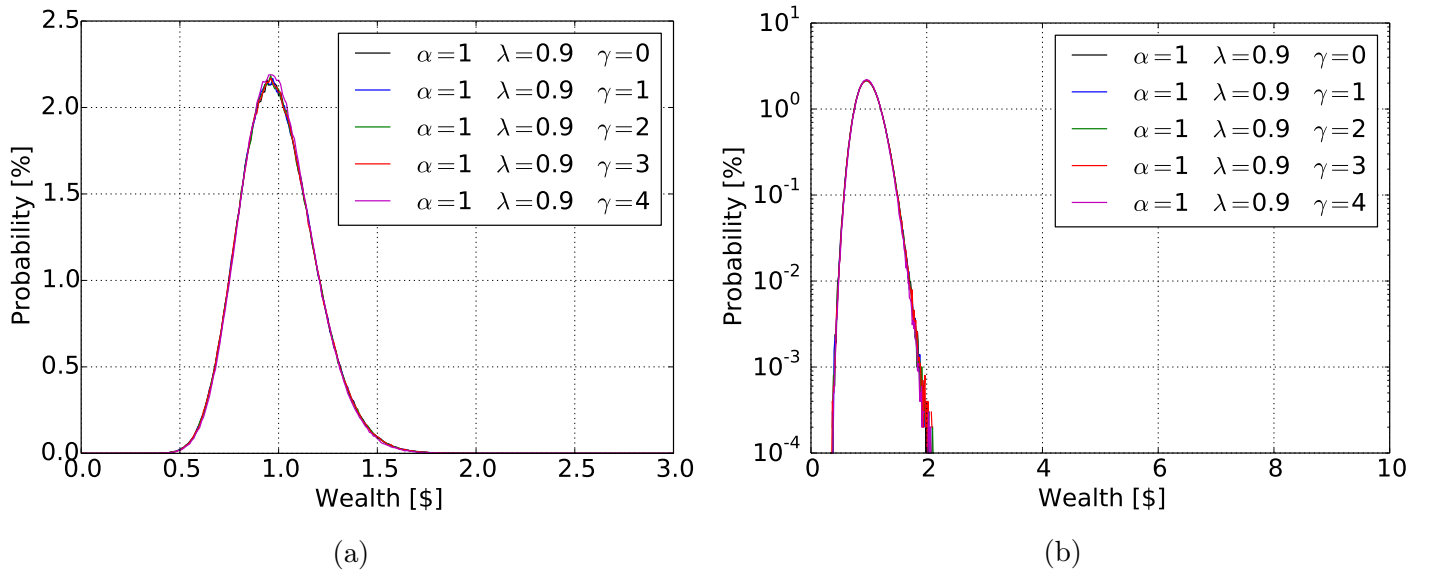
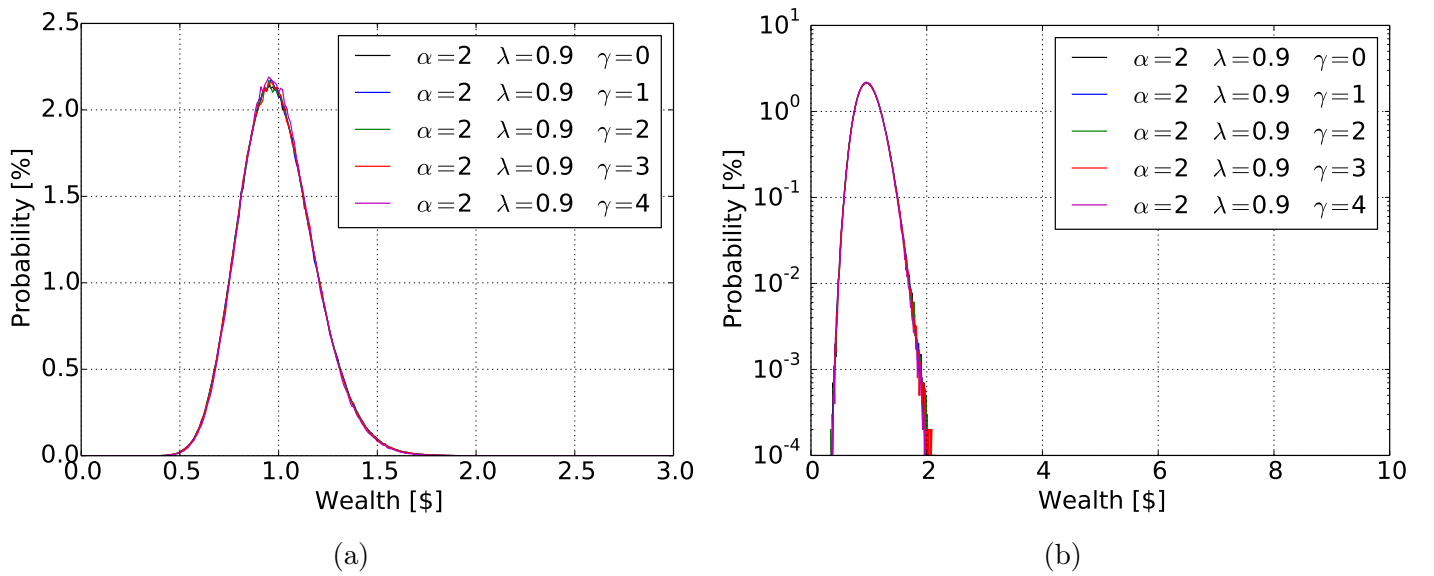


Figure 10: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

5.4 5e

Figure 11: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .Figure 12: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

Figure 13: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .Figure 14: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

Figure 15: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .Figure 16: a) Shows how E behaves around T_C b) Shows how $|M|$ develops near T_C .

6 Conclusion

7 References

References

- [1] Sanchari Goswami and Parongama Sen. “Agent based models for wealth distribution with preference in interaction”. In: *Physica A: Statistical Mechanics and its Applications* 415.Supplement C (2014), pp. 514–524. ISSN: 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2014.08.018>. URL: <http://www.sciencedirect.com/science/article/pii/S0378437114006967>.
- [2] Peter Hudomiet, Gabor Kezdi, and Robert J. Willis. “Stock market crash and expectations of American households”. In: *Journal of Applied Econometrics* 26.3 (2011), pp. 393–415. ISSN: 1099-1255. DOI: 10.1002/jae.1226. URL: <http://dx.doi.org/10.1002/jae.1226>.
- [3] F. Rouge. Lausanne. “PARETO, V. (1896)”. In: (1896).
- [4] Marco Patriarca, Anirban Chakraborti, and Kimmo Kaski. “Gibbs versus non-Gibbs distributions in money dynamics”. In: *Physica A: Statistical Mechanics and its Applications* 340.1 (2004). News and Expectations in Thermostatistics, pp. 334–339. ISSN: 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2004.04.024>. URL: <http://www.sciencedirect.com/science/article/pii/S0378437104004327>.
- [5] “Editorial Board”. In: *Journal of Financial Markets* 36 (2017), IFC–. ISSN: 1386-4181. DOI: [https://doi.org/10.1016/S1386-4181\(17\)30290-2](https://doi.org/10.1016/S1386-4181(17)30290-2). URL: <https://www.sciencedirect.com/science/article/pii/S1386418117302902>.

8 Appendix