

# Monte Carlo Modeling of Transactions

Ingrid Marie Kjelseth, Elisabeth Strøm, Renate Mauland

December 4, 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Method</b>	<b>3</b>
2.1	Basic model . . . . .	3
2.2	Including saving . . . . .	3
2.3	Preferences for transactions . . . . .	4
2.3.1	Wealth . . . . .	4
2.3.2	Former transactions . . . . .	4
2.4	Algorithm . . . . .	5
<b>3</b>	<b>Implementation</b>	<b>5</b>
<b>4</b>	<b>Results</b>	<b>5</b>
<b>5</b>	<b>Discussion</b>	<b>5</b>
<b>6</b>	<b>Conclusion</b>	<b>5</b>

## Abstract

# 1 Introduction

# 2 Method

Here we present the methods used to achieve the results in Section ????. We will use Monte Carlo methods to simulate financial transactions among agents. In our model we have a total of  $N$  agents, where a money transaction takes place between a pair of agents,  $i$  and  $j$ , at a time. Each agent starts of with the same amount of money,  $m_0 = 100$ , and the total amount of money is  $m = \sum_i^N m_0$ . Money is conserved in the transaction, so

$$m'_i + m_i = m'_j + m_j, \quad (1)$$

where  $m_i$  and  $m_j$  is the amount of money of agent  $i$  and  $j$ , respectively, and the apostrophe denotes the wealth of an agent after a transaction has taken place.

## 2.1 Basic model

Assuming now a simple model, where the amount of money that changes hands between two agents is random. The wealth of each agent,  $i$  and  $j$ , after the transaction has taken place is given by

$$m'_i = \epsilon(m_i + m_j) \quad m'_j = (1 - \epsilon)(m_i + m_j), \quad (2)$$

, where  $\epsilon$  is a random number between zero and one,  $\epsilon \in [0, 1]$ , meaning an agent can not be in debt to another.

Due to the law of conservation defined in Equation 1, one can show that the system eventually reaches an equilibrium state given by a Gibbs distribution,

$$w_m = \beta e^{-\beta m}. \quad (3)$$

Here,  $\beta = 1/\langle m \rangle$ , and  $\langle m \rangle$  is the average money, defined as

$$\langle m \rangle = \sum_i^N \frac{m_i}{N} = m_0, \quad (4)$$

the initial wealth of each agent. We can compare the Gibbs distribution with the one we get through running several Monte Carlo simulations. We have that  $N = 500$  and so the total money is  $m = 50\,000$ .

## 2.2 Including saving

To make our model more realistic, we will add the ability of an agent to save their money. We do this by adding a criterion stating that an agent can save a fraction  $\lambda$  of their money. This means that the money that can be used in a transaction between agents  $i$  and  $j$  is  $(1 - \lambda)(m_i + m_j)$ .

The individual wealth after a transaction can be written as

$$m'_i = m_i + \delta m \quad m'_j = m_j - \delta m, \quad (5)$$

where we have introduced the change in money  $\delta m$  as

$$\delta m = (1 - \lambda)(\epsilon m_j - (1 - \epsilon)m_i). \quad (6)$$

We will run our computations for this case with  $\lambda = 0.25, 0.5$  and  $0.9$ . From our calculations we can extract the probability of an agent having a certain amount of money. This probability is no longer given by a Gibbs distribution, but is well fitted by the function

$$w_m = P_n(x) = a_n x^{n-1} e^{-nx} \quad (7)$$

from Patriarca et al. (2004). Here we have introduced the variables

$$x = \frac{m}{\langle m \rangle}, \quad \text{and} \quad n(\lambda) = 1 + \frac{3\lambda}{1-\lambda}, \quad a_n = \frac{n^n}{\Gamma(n)}, \quad (8)$$

where  $\Gamma(n)$  is the Gamma function. The number of agents we have in our simulation is still  $N = 500$  and  $m = 50\,000$ .

## 2.3 Preferences for transactions

So far, any agent can interact with any agent with an equal probability. In reality this is seldom the case, people have preferences for who they want to conduct business with. There are different ways of simulating this, but we will run our simulations based on two assumptions: People with the same amount of wealth are more likely to interact with one another; People who have previously done transactions are more probable of doing so again with the same person. Note that we do not include the possibility of past transactions having been a negative experience for our agents, thus impacting the probability of repeat interactions.

### 2.3.1 Wealth

First we include the first mentioned assumption; making the probability of two agents interacting a function of their respective wealth. We define this probability as

$$p_{ij} = 2|m_i - m_j|^{-\alpha}. \quad (9)$$

We see that if the agents possesses a similar wealth, the likelihood of them interacting increases. We will run this simulation for various  $\alpha \in [0, 2]$ . A comparison will be made between the tail end of our final distribution of wealth and the Pareto distribution, in an attempt to reproduce Figure 1 from Goswami and Sen (2014). The Pareto distribution is given by

$$w_m = A m^{-(1+\alpha)}, \quad (10)$$

where the proportionality constant  $A$  is decided by calculating the slope of the curve of our simulated wealth distribution. We intend to do this analysis with and without saving, meaning we will put  $\lambda = 0$  and  $\lambda > 0$ . The simulations will be run for  $N = 500$ , and  $N = 1000$ .

### 2.3.2 Former transactions

Next we include how two agents having interacted before increases the probability of them doing so again. We simply multiply the previous likelihood with a new factor:

$$p_{ij} = 2|m_i - m_j|^{-\alpha} \left( \frac{c_{ij} + 1}{\max(\mathbf{C}) + 1} \right)^\gamma, \quad (11)$$

where  $c_{ij}$  is the number of previous interactions between agents  $i$  and  $j$ , which is an element of the  $\mathbf{C}$  matrix, which contain every interaction between all pair of agents. We add the 1 for the case where they have not interacted before, so that it is still possible for them to do so.

We will compare this result with Figure 5 and 6 in Goswami and Sen (2014), meaning we will do this analysis for  $N = 1000$ ,  $\lambda = 0$ ,  $\alpha = 1$  and  $\alpha = 2$  using  $\gamma \in [1, 4]$ . We will also do a case with saving, putting  $\lambda = \dots$

## 2.4 Algorithm

To model the financial transactions, we will run a thousand Monte Carlo cycles. For each cycle we first initialize the transaction matrix,  $\mathbf{C}$ , which keep tabs over how many transactions has occurred between agents  $i$  and  $j$ . We also initialize a cash array, which has a length equal to the number of agents  $N$ . The money of each agent is added here for each transaction. The matrix is initially filled with zeros, while the cash array is filled with the initial sum  $m_0$ . The next thing that happens is the transactions them selves. The algorithm for this looks as follows;

1. We pick out two random numbers between 0 and  $N - 1$ . These are the agents that might do a transaction.
2. We calculate the probability  $p_{ij}$  Of the two agents performing a transaction. If  $\alpha$  and  $\gamma$  is zero, this will always happen.
3. We pick out a random number between 0 and 1 .If this number is lower than  $p_{ij}$ , they will perform a transaction. In which case;
  - Pick a random number  $\epsilon$  from a uniform distribution.
  - Calculate the new wealth of agent  $i$  and  $j$  as defined by Equation 5 and 6.
  - Add a number in spot  $\mathbf{C}_{ij}$  in the transaction matrix to indicate that a successful transaction has occurred between Agent  $i$  and Agent  $j$ .
4. Repeat steps 1 through 4 as many times as has been decided. We repeat the above between  $10^5$  to  $10^7$  times, depending on the values of the parameters  $\lambda$ ,  $\alpha$ , and  $\gamma$ .

Doing this for every spin in our lattice, constitutes one Monte Carlo cycle, of which we will be doing several. We expect that for many MC cycles, such as  $10^7$  of them, the numerically computed expectation values divided by the number of MC cycles will converge toward their actual analytical values.

This is the idea behind a Markov chain: There are two possible outcomes to each experiment, and there is a probability for each outcome, which depend on the previous outcome of the previous experiment. As time goes on, or rather, as the number of MC cycles increases, the system reaches an equilibrium state. In our case, the two outcomes are that a spin is flipped, or that it is not flipped, and the probability for either depend on the energy of the previous flip, or rather the previous “experiment”.

## 3 Implementation

## 4 Results

## 5 Discussion

## 6 Conclusion

## References

- Sanchari Goswami and Parongama Sen. Agent based models for wealth distribution with preference in interaction. *Physica A: Statistical Mechanics and its Applications*, 415(Supplement C):514 – 524, 2014. ISSN 0378-4371. doi: <https://doi.org/10.1016/j.physa.2014.08.018>. URL <http://www.sciencedirect.com/science/article/pii/S0378437114006967>.
- Marco Patriarca, Anirban Chakraborti, and Kimmo Kaski. Gibbs versus non-gibbs distributions in money dynamics. *Physica A: Statistical Mechanics and its Applications*, 340(1):334 – 339, 2004. ISSN 0378-4371. doi: <https://doi.org/10.1016/j.physa.2004.04.024>. URL <http://www.sciencedirect.com/science/article/pii/S0378437104004327>. News and Expectations in Thermostatistics.