Project 4, deadline May 2

Stock Market model

Spring semester 2018

Theoretical background and description of the system

The aim of this project is to simulate financial transactions among financial agents using Monte Carlo methods. The final goal is to extract a distribution of income as function of the income m. From Pareto's work (V. Pareto, 1897) it is known from empirical studies that the higher end of the distribution of money follows a distribution

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

with $\alpha \in [1, 2]$. We will here follow the analysis made by Patriarca and collaborators.

Here we will study numerically the relation between the micro-dynamic relations among financial agents and the resulting macroscopic money distribution.

We assume we have N agents that exchange money in pairs (i, j). We assume also that all agents start with the same amount of money $m_0 > 0$. At a given 'time step', we choose randomly a pair of agents (i, j) and let a transaction take place. This means that agent i's money m_i changes to m_i' and similarly we have $m_j \rightarrow m_j'$. Money is conserved during a transaction, meaning that

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

The change is done via a random reassignement (a random number) ϵ, meaning that

$$m_i' = \epsilon(m_i + m_j),$$

leading to

$$m_j' = (1 - \epsilon)(m_i + m_j).$$

The number ϵ is extracted from a uniform distribution. In this simple model, no agents are left with a debt, that is $m \geq 0$. Due to the conservation law above, one can show that the system relaxes toward an equilibrium state given by a Gibbs distribution

$$w_m = \beta \exp(-\beta m),$$

with

$$\beta = \frac{1}{\langle m \rangle},$$

and $\langle m \rangle = \sum_i m_i/N = m_0$, the average money. It means that after equilibrium has been reached that the majority of agents is left with a small number of money, while the number of richest agents, those with m larger than a specific value m', exponentially decreases with m'.

We assume that we have $N = 500$ agents. In each simulation, we need a sufficiently large number of transactions, say 10^7. Our aim is find the final equilibrium distribution w_m. In order to do that we would need several runs of the above simulations, at least $10^3 - 10^4$ runs (experiments).

Project 4a): Simulation of Transactions. Your task is to first set up an algorithm which simulates the above transactions with an initial amount m_0. The challenge here is to figure out a Monte Carlo simulation based on the above equations. You will in particular need to make an algorithm which sets up a histogram as function of m. This histogram contains the number of times a value m is registered and represents $w_m \Delta m$. You will need to set up a value for the interval Δm (typically $0.01 - 0.05$). That means you need to account for the number of times you register an income in the interval $m, m + \Delta m$. The number of times you register this income, represents the value that enters the histogram. You will also need to find a criterion for when the equilibrium situation has been reached.

Project 4b): Recognizing the distribution. Make thereafter a plot of $\log(w_m)$ as function of m and see if you get a straight line. Comment the result.

Project 4c): Transactions and savings. We can then change our model to allow for a saving criterion, meaning that the agents save a fraction λ of the money they have before the transaction is made. The final distribution will then no longer be given by Gibbs distribution. It could also include a taxation on financial transactions.

The conservation law of Eq. (1) holds, but the money to be shared in a transaction between agent i and agent j is now $(1 - \lambda)(m_i + m_j)$. This means that we have

$$m_i' = \lambda m_i + \epsilon(1 - \lambda)(m_i + m_j),$$

and

$$m_j' = \lambda m_j + (1 - \epsilon)(1 - \lambda)(m_i + m_j),$$

which can be written as

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

and

$$m'_j = m_j - \delta m,$$

with

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

showing how money is conserved during a transaction. Select values of $\lambda = 0.25, 0.5$ and $\lambda = 0.9$ and try to extract the corresponding equilibrium distributions and compare these with the Gibbs distribution. Comment your results. Extract a parametrization of the above curves, see for example Patriarca and collaborators and see if you can parametrize the high-end tails of the distributions in terms of power laws. Comment your results.

Project 4d): Nearest neighbor interactions. In the rest of this project we will follow the work of Goswami and Sen. In the studies above the agents were selected randomly, irrespective of whether we allowed for saving or not during a transaction. What is often observed is that various agents tend to make preferences for for whom to interact with. We will now study the evolution of the distribution of wealth w_m by assuming that there is a likelihood

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

for an interaction between agents i and j with respective wealths m_i and m_j. The parameter $\alpha > 0$. For $\alpha = 0$ we recover our model from part 5a). Perform the same analysis as previously with $N = 500$ as well as with $N = 1000$ agents and study the distribution of wealth for $\alpha = 0.5$, $\alpha = 1.0$, $\alpha = 1.5$ and $\alpha = 2.0$. You should try to reproduce Figure 1 of Goswami and Sen. Extract the tail of the distribution and see if it follows a Pareto distribution

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

What happens if $\alpha \gg 1$?

Perform the analysis with and without a saving λ on each transaction and comment your results.

Project 4e): Nearest neighbors and former transactions. We add to the previous probability the possibility that two agents who interact have performed similar transactions earlier. That is, in addition to being financially close, we assume that the likelihood for interacting increases if two agents have interacted earlier. We add this feature by modifying the previous likelihood to

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

where c_{ij} represents the number of previous interactions that have taken place between i and j. The factor 1 is added in order to ensure that if they have not interacted earlier they can still interact. Perform similar studies as above with $N = 1000$, $\alpha = 1.0$ and $\alpha = 2.0$ using $\gamma = 0.0, 1.0, 2.0, 3.0$ and 4.0. Plot the wealth distributions for these cases and try to extract eventual power law tails with and without a saving λ in each transaction. Comment your results and compare them with figures 5 and 6 of Goswami and Sen.

Finally, (this part is optional) if you have time, which features would you add to these models in order to make them even more realistic?

Background literature

- V. Pareto, Cours d'economie politique, Lausanne, 1897.

- M. Patriarca, A. Chakraborti, K. Kaski, Physica A **340**, 334 (2004).

- S. Goswami and P. Sen, Physica A **415**, 514 (2014).

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.

- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.

- Include the source code of your program. Comment your program properly.

- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.

- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.

- Try to evaluate the reliabilty and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.

- Try to give an interpretation of you results in your answers to the problems.

- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.

- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Use your github repository to upload your report. Indicate where the report is by creating for example a **Report** folder. Please send us as soon as possible your github username.

- Place your programs in a folder called for example **Programs** or **src**, in order to indicate where your programs are. You can use a README file to tell us how your github folders are organized.

- In your git repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parameters.

- In this and all later projects, you should include tests (for example unit tests) of your code(s).

- Comments from us on your projects, with score and detailed feedback will be emailed to you.

Finally, we encourage you to work two and two together. Optimal working groups consist of 2-3 students. You can then hand in a common report.