University of Zurich UZH

Department of Economics

Mastering Agent-Based Economics

Master's Thesis in Economics – November 2015

Luzius Meisser (mail@luziusmeisser.com)
Supervisors: Dr. Johannes Brumm, Prof. Felix Kübler, Department of Banking and Finance
Advisor Software Engineering: Prof. Abraham Bernstein, Department of Informatics

Online: master.agentecon.com/thesis.pdf

To Verena, Nathanael, and Sienna

Abstract

An agent-based production economy with saving consumers and a stock market is incrementally built, applying methods from modern software engineering and benchmarking outcomes with classic equilibrium results. A suite of complementary tools is key to attaining high accuracy and robustness: exponential search for price finding, sensor prices to separate information exploitation from information exploration, a stability ranking of seemingly identical firm decision heuristics, a decentralized variant of Walras' tâtonnement process to accelerate convergence, and reincarnating agents to find self-confirming equilibria in configurations with anticipated shocks. The production economy is extended by listing the dividend-bearing shares of firms on a stock market, where overlapping generations of mortal consumers save for retirement. Market makers provide liquidity and are shown to qualitatively impact price formation. Further adding fundamentalist traders that are listed companies themselves can lead to self-reinforcing, chaotic dynamics driven by circular ownership structures.

Acknowledgements

I wholeheartedly thank Johannes Brumm and Prof. Felix Kübler for providing me with the freedom to dedicate this thesis to my topic of choice. I am grateful to Johannes Brumm and to Prof. Abraham Bernstein for their valuable inputs and support. I am also thankful for the comments by various participants of the CEF 2015 conference – most notably Ulrich Wolffgang –, and I thank Friedrich Kreuser for his comments and help with the revised version of our joint paper. I am thankful to Alberto Russo and Matthias Lengnick for giving me access to the source code of their models, as well as to all the authors whose models are publicly available.

In particular, I am deeply grateful to my lovely wife for her support, as well as to our children for reminding me of the real life every now and then. I am also deeply grateful to my parents and parents-in-law for regurarly looking after them, allowing me to fully focus on this thesis. Lastly, I would like to mention Gizmo the cat, who did not help much, but patiently watched me typing for countless hours.

Contents

List of Figures

List of Tables

1. Introduction

"In using systems of adaptive agents to create a 'guess', we are counting on the tendencies of these systems of adaptive agents, as plodding as they are, eventually to find their ways to equilibria that we economists (who, after all, made up the model!) have difficulty finding."
- Thomas Sargent (1993)

Agent-based modeling is inherently constructive, enabling the creation of macroeconomic models from the bottom up. (Tesfatsion, 2006, Epstein, 2006, Gatti et al., 2011) Its atomic building blocks are individual, stateful, interacting agents. If done right, these agents collectively behave in ways that cannot be directly attributed to the decisions of individuals, making the whole greater than the sum of its parts. Simulated ants, for example, can exhibit complex collective behavior despite each individual ant only following trivial rules. (Macal and North, 2005) Similarly, the goal of agent-based economics should be to create simulations consisting of relatively simple agents that collectively exhibit rich behavior with the overall outcome naturally emerging as a result of their interactions. The simulation is pushed towards its equilibrium by Adam Smith' famous invisible hand.

The difference between the use of information in agent-based models and the use of information in equation-based models[1] resembles the difference between decentralized and centralized planning in *The Use of Knowledge in Society* by Hayek (1945). With equation-based approaches, a model consists of a monolithic equation system. However, just like Hayek's central planner, such models struggle at incorporating diverse knowledge and are usually based on radically simplified assumptions. Farmer and Foley (2009) criticize the most popular flavor of equation-based models as follows: "Even if rational expectations are a reasonable model of human behaviour, the mathematical machinery is cumbersome and requires drastic simplifications to get tractable results. The equilibrium models that were developed, such as those used by the US Federal Reserve, by necessity stripped away most of the structure of a real economy."

In contrast, agent-based models allow to handle much higher complexity. No one, not even the architect of the model, needs the mental capacity to ever grasp the model as a whole in all its detail. Instead, individual agents can be programmed and tested one at a time, each of them acting on its own local informations and beliefs – just like Hayek's individuals in the decentralized economy. The glue between them is the market and its prices. What Hayek (1945) says about the real world also applies to agent-based models: "In a system in which the knowledge of the relevant facts is dispersed among many people, prices can act to coordinate the separate actions of different people."

While agent-based models allow for high complexity, they also suffer from a number of shortcomings and restrictions. Time is directional, many quantities discrete, and aggregate variables cannot be exogenously set. However, the primary challenge is their often chaotic nature. Seemingly irrelevant details and subtle programming errors can decidedly shape the aggregate outcome. Again Farmer and Foley (2009): "An attempt to model all the details of a realistic problem can rapidly lead to a complicated simulation where it is difficult to determine what causes what. To make agent-based modelling useful we must proceed systematically, avoiding arbitrary assumptions, carefully grounding and testing each piece of the model against reality and introducing additional complexity only when it is needed."

While the importance of testing is undisputed, testing against well-known, analytically verifiable results can be preferreble to testing against reality. It provides a faster, simpler, and more precise benchmark. In this regard, I follow the footsteps of authors such as Brock and Hommes (1998), Bullard and

[1] The term *equation-based model* is adopted from Parunak et al. (1998) who compare agent-based to equation-based models in the context of supply chains. It broadly refers to mathemathical models. Some agent-based models can also be described and solved mathematically. However, in that case, they are usually referred to as heterogenous agent models instead.

Duffy (1999), Gintis (2007), or LeBaron (2001), who explicitely recommends benchmarking agent-based models against classic equilibrium results.

The authors that advocate constructing agent-based models close to classic equilibrium models can be devided into two subgroups. The majority sees agent-based modeling as a method of qualitatively extending the scope of what can be modeled. Often, endogenous learning is emphasized - something that rational expectations models cannot cover by definition as rational expectations require the agents to know the rules that govern the world from day one. A small minority sees agent-based models less spectacularily as a numerical computation tool, capable of finding solutions where other methods fail. For example, Wright (1995) applies a genetic algorithm to let the agents of his model find an equilibrium for him, without caring about the evolutionary dynamics. While I believe that agent-based models can tremendously extend the scope of what can be modeled, I also believe that one should start by replicating existing results. Thereby, this relatively new and complex tool can be put on verifiable methodological foundations, from where further exploration is possible with more confidence.

The basic version of the presented model consists of consumers and firms in a sequence of Arrow-Debreu spot markets with endogenous price finding. It converges towards the pareto-efficient equilibrium. Its extended version contains saving consumers with overlapping generations and a fully-functional stock market. Unlike other models, it has no banks, no government, no long-term contracts, no artificial intelligence, and only a trivial fully connected topology. Examples of more complex models that include some of these features are the Eurace project by Deissenberg et al. (2008), the family of models by Gatti et al. (2011), the Jamel framework by Seppecher (2012), and the crisis economics initiative[2] driven by visionary publications such as Cincotti et al. (2012) or Farmer et al. (2012). Due to their complexity, it is non-trivial to rigorously test these models. They are usually only qualitatively verified by means of stylized facts and more recently by ensuring stock-flow consistency (Caiani et al., 2015). Stock-flow consistency requires that stocks of goods and financial assets must be correctly accounted for. The presented model fulfills this invariant, whereas many earlier ones – in particular financial market models such as the seminal work of Lux and Marchesi (1999) – do not.

Having enforcable invariants and an accurate benchmark enables a development style known as *fail fast*. It significantly improves the speed at which errors in the model are detected and thus also can be addressed. (Shore, 2004) Other discussed methods from software engineering that help in building agent-based models are object-oriented programming, version control systems, and the consequences of seeing the code as the model. Underscoring their importance, the applied methods have their own dedicated chapter *Methodology*.

The agent-based simulation of the Stolper-Samuelson effect presented at CEF 2015 (Meisser and Kreuser, 2015) serves as a starting point for the production economy described in chapter *Production Economy*. While this earlier model already included exponential search and could replicate the Stolper-Samuelson effect, it is completely revised. It is made pareto-efficient by making firms profit-seeking and distributing dividends, and the input synchronization problem is addressed by the introduction of *sensor prices*. Furthermore, causal loop diagrams are used to illustrate and analyze the system's dynamics. These improvements have been incorporated in a revised paper attached as appendix C and submitted to *Computational Economics*.[3] Considerations regarding the dynamics of the production economy are moved to their own chapter *Model Dynamics*, which also contains a ranking of seemingly equivalent dividend schedules as well as a descentralized variant of Walras' tâtonnement process to accelerate convergence.

[2] Resulting publications can be found on crisis-economics.eu and the model itself on github.com/crisis-economics/CRISIS

[3] Not reviewed yet at the time of writing. A declaration of authorship can be found along with the paper in appendix C.

While agent-based models generally do not permit rational expectations, I propose *Reincarnating Agents* as an exogenous learning method to find self-confirming equilibria over the course of multiple simulation runs. Such equilibria can coincide with rational expectations equilibria under suitable conditions. This method enables the introduction of mortal consumer agents with savings and consumptions that act in line with the equivalent rational expectations model.

Finally, the model is extended further in chapter *Stock Market* by allowing the dividend-bearing shares of firms to be publicly traded. Liquidity is provided by profit-seeking market makers, and an attempt is made to drive stock prices towards their fundamental value by the introduction of investment fund agents with a value strategy. Allowing these fundamentalists to buy each other's shares can lead to booms and busts that share some of the statistical properties of real-world markets and that can destabilize the underlying economy. They are driven by the emergence of circular ownership structures and the amplifying effect of share buybacks on mispricings.

2. Methodology

"Programming frees us to adapt the tool to the problem rather than the problem to the tool."
- Leigh Tesfatsion (2006)

The choice of tools visibly impacts the end-result. An artists preference for pencil over paintbruch leads to a different piece of art, even when the depicted motif stays the same. My tools of choice stem from my experience in software engineering, a field that is concerned with the reliable construction of complex systems. It offers proven toolkits to address some of the pitfalls of constructing agent-based economics.

This section starts with the basic abstractions of agent-based modeling and then gradually moves to increasingly technical topics, ending with the software architecture. I hope others following a similar path can learn from what worked well and what worked not so well for me. A good methodology speeds up development, supports testing, improves readability and accessibility, helps focusing on the relevant, and makes playing with the model a joyful experience. Readers that only care about the presented agent-based model itself can safely skip this chapter.

2.1. Agent-Based and Object-Oriented

The first object-oriented language was Simula, invented by Dahl and Nygaard (1966) to provide humans with an intuitive abstraction to program simulations. Incidentally, objects also are an excellent abstraction to manage complexity by enscapsulating separate concerns, making object-orientation the most popular programming style by far today.[4]

As Tesfatsion (2006) and others point out, object-orientation resembles agent-based modeling.[5] Individual agents act in accordance with private beliefs, which they update by observing local information. The agent-based approach intuitively is a suitable way to write models consisting of individuals. It allows to escape the shortcoings of aggregation, which usually is a necessity in equation-based models.[6] Incidentally, agents also are an excellent abstraction to manage complexity by ensapsulating separate concerns - just like object-orientation.

Complexity is reduced two-fold. First, a clean separation of concerns allows to implement and test each agent individually, without having to care much about the rest of the program, thereby significantly reducing the mental load of the programmer. Encapsulation makes the whole system much less fragile than monolithic systems with globally visible variables, which cannot guarantee that a given

[4] An earlier paradigm is *procedural programming*, whose structure comes closer to the ODD protocol proposed by Grimm et al. (2006) and discussed in subsequent 2.2. A newer paradigm that is especially suited for concurrent programming is *functional programming*, with Julia (*julialang.org/*) being an example of a functional language designed for scientific computing. The traditional object-oriented programming languages such as C# and Java also tend to adopt more and more features from functional programming with each revision. Of these three paradigms, object-oriented programming comes closest to agent-based model structures. Language that natively support *active objects* - for example Active Oberon developed at ETH Zurich - might come even closer, but are not commercially established.

[5] In an interview with the Rolling Stone (rollingstone.com/culture/news/steve-jobs-in-1994-the-rolling-stone-interview-20110117), Steve Jobs defines objects in a way that could easily apply to agents as well: "If I'm your laundry object, you can give me your dirty clothes and send me a message that says, 'Can you get my clothes laundered, please.' I happen to know where the best laundry place in San Francisco is. And I speak English, and I have dollars in my pockets. So I go out and hail a taxicab and tell the driver to take me to this place in San Francisco. I go get your clothes laundered, I jump back in the cab, I get back here. I give you your clean clothes and say, 'Here are your clean clothes.' You have no idea how I did that. You have no knowledge of the laundry place. Maybe you speak French, and you can't even hail a taxi. You can't pay for one, you don't have dollars in your pocket. Yet I knew how to do all of that. And you didn't have to know any of it. All that complexity was hidden inside of me, and we were able to interact at a very high level of abstraction. That's what objects are. They encapsulate complexity, and the interfaces to that complexity are high level."

[6] Ugly properties of aggregation have already been proven early on, an example being the Sonnenschein-Mantel-Debreu theorem discussed in Sonnenschein (1973). Also the works of the latest Nobel laureate Angus Deaton are largely concerned with the complications of aggregation, for example in Deaton and Zaidi (2002).

change at one end of the model does not adversely impact something seemingly unrelated at the other end. Second, it is often much easier to solve many small problems than to solve one big problem. Thus, finding a valid way to split a large problem into small digestable units is an excellent solution strategy, sometimes referred to as *Divide et Impera*. (Gutknecht and Hromkovic, 2010) Normally, it is much easier to optimize the behavior of an individual agent instead of the whole system, bringing us back to the initially discussed thoughts of Hayek (1945).

The opposite of object-orientation is to separate data from functions. Relational databases and all other abstractions that rely on a matrix-like representation of data do so. Consequently, there are always frictions when trying to make use of both paradigms at the same time.

2.2. The Code is the Model

"An idea is nothing, its implementation everything."
- Alexander Kronrod (Landis et al., 2002)

Equation-based models are fundamentally mathematical. They are represented by a system of equations that are solved analytically or numerically. Agent-based models are fundamentally algorithmic.[7] They are represented by a number of agents with distinct state and behavior. The only way to solve them is by simulation. Thus, the cleanest way to specify equation-based models is to use mathematical terms, and the cleanest way to specify agent-based models is to use source code.

Rhett Jones visited New York in 2015 with a tourist guide from 1997. He found nine out of ten recommended clubs closed and half the book stores gone.[8] In software engineering, there is often a similar discrepancy between specification documents and the actual software. This leads to a heated debate about what the design of a software actually is. Is the design in the specification documents, or is it in the source code? Proponents of agile software development argue for the latter.[9]

In theory, written specifications can be kept up to date. In practice, they are often outdated within days after the implementation phase of traditional development processes has started. If one wants to be certain about what a program actually does, its code often is the only reliable source.[10] Thus, Reeves (2005) concludes: "In software development, the design document is a source code listing."

Agile software development methodologies are built upon this insight. Typically, they come with taglines like *"the code is the design"* by Reeves (2005) or *"the code is the documentation"*.[11] Similarly, I argue that *the code is the model*. This is a pragmatist view stemming from the observation that I created and refined my model iteratively while programming, as opposed to writing down its specification on paper first. Seeing the code as the model implies that editing and adapting the model is done by editing and adapting its source code. The model as specified by the source code becomes the primary deliverable, with accompanying papers focusing on the documentation of specific insights.

Unfortunately, this is far from being the consensus view. Often, attempts are made to specify agent-based models in natural language, with a few mathematical equations mixed in where applicable.

[7] In the words of Salle et al. (2013): "Agent-based models are sequential by nature, and the sequence of events has to be described step by step."

[8] See hopesandfears.com/hopes/now/experiment/168771-90s-tourist-guide-nyc.

[9] The Agile Manifesto lists the principles of agile development.

[10] As an example, I often looked at source code of other models to see which of the price adaption heuristic from section 3.3 they actually use, a detail the relevant papers often are silent or unclear about.

[11] An insightful comment by Martin Fowler, an influential software design expert, can also be found on martin-fowler.com/bliki/CodeAsDocumentation.html

Grimm et al. (2006) note that this leads to unsatisfactory model descriptions, for example "not including enough detail of the model's schedule to allow the model to be re-implemented". As a remedy, they propose the ODD protocol to describe agent-based models. They later update and refine the protocol in Grimm et al. (2010), noting that they had often been criticized for proposing variable tables that are presented seperately from the functions that operate on these variables. They believe that: "Once readers know the full set of (low-level) state variables, they have a clear idea of the model's structure and resolution." By doing so, Grimm et al. tear apart the object-oriented design, as they note themselves.

In contrast, object-orientation organizes the program hierarchically, allowing the reader to focus on the variables and functions of interest. Nonetheless, one must assume that the average reader of an article is neither interested nor trained in reading source code. Thus, the article must provide all relevant abstractions to understand the model as well as all relevant results. The target audience of the source code are those who want to reproduce results or find out about a very specific detail.

2.3. Version Control

Seeing the code as the model allows the author (and inclined readers) to fully leverage the powerful tools available for software development. Software engineering is probably the most advanced discipline in collaboratively creating complex systems. Much more resources have been invested into creating good software development tools than into creating good economic modeling tools. One category of such tools are version control systems. Version control systems store the complete history of all code changes, serve as source code repository, and enable collaboration with in some cases thousands of programmers concurrently working on the same code base. Maybe, one day, there are large-scale economic models with dozens of economists from all over the world concurrently improving and refining them?

My version control system of choice is Git, which was recently discussed in the context of computational economics by Bruno (2015). Git is an open standard, with many competing repositories and clients available. A popular choice is to use Github (github.com) as repository and SourceTree (sourcetreeapp.com) as client. Figure 1 shows a screenshot of browsing code written by Steve Phelps, one of the authors that publish their source code in a public repository.[12] Also available in a public Git repository is the model of Wolffgang (2015),[13] and that of the crisis project.[14]

The publisher is rarely the first choice to host source code, even though they usually offer to host supplementary material. This is probably owed to the static nature of the classic journal papers and inexistent integration with version control tools such as Git. Sometimes, links to web resources do not even work.[15] The established academic processes do not seem to encourage the publishing of source code and the maintenance web links.

Recently, Chang and Phillip (2015) tried to reproduce the results of 67 economics papers published in a selection of 13 reputable journals. They could only replicate 33% of them on their own, and 43% with the authors' assistance. The primary reason for a failure to replicate the results was missing software or data – even for journals that in theory have a policy of requiring source code and data. They recommend making the provision of source code and data a strict condition for publication in all journals.

[12] JABM Git repository: github.com/phelps-sg/jabm

[13] Git repository of Wolffgang's Computational Economy: github.com/uwol/ComputationalEconomy

[14] Crisis Git repository: github.com/crisis-economics/CRISIS

[15] For example, Gatti et al. (2011) write: "For further information, please visit the following link: www.springer.com/series/9601." The link does not work.

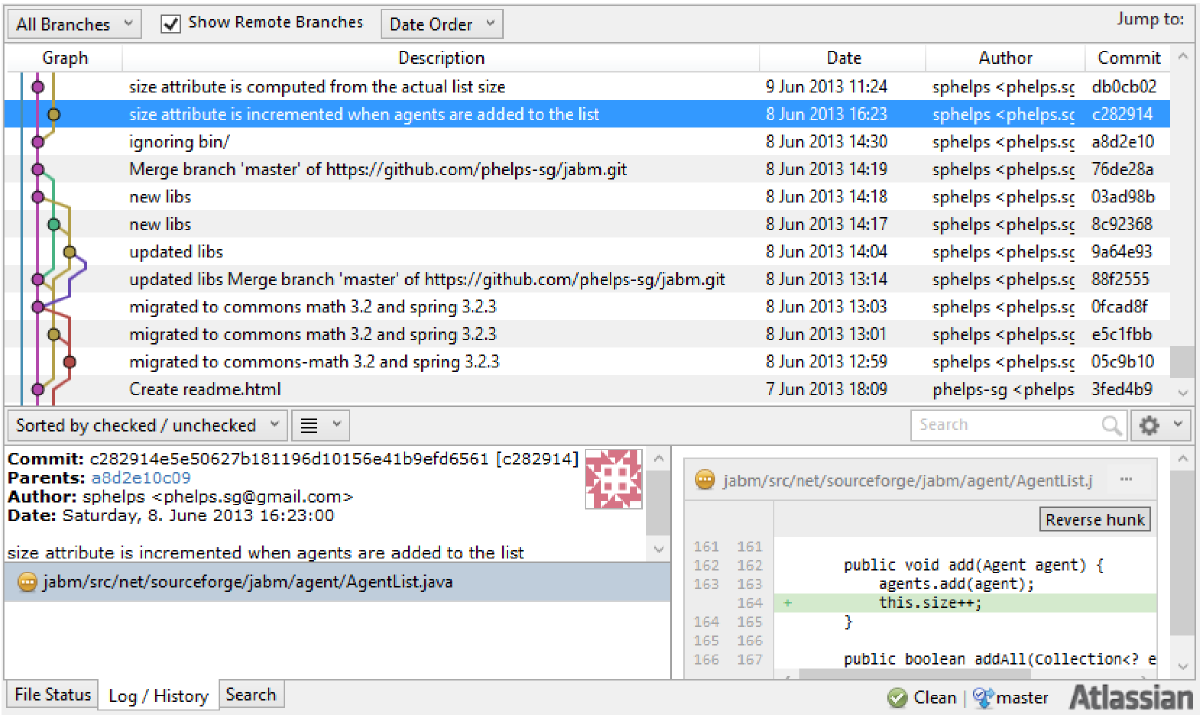


Figure 1: A screenshot of the Git client SourceTree, showing changes Steve Phelps made to the Java Agent-Based Modeling library (jabm.sourceforge.net), which was used to produce the results presented in Caiani et al. (2014) and Caiani et al. (2015). The selected commit to the yellow branch with fingerprint c282914 apparently fixed the size counter of a list of agents named AgentList. Given well-commented and frequent commits, Git also is a useful lab journal.

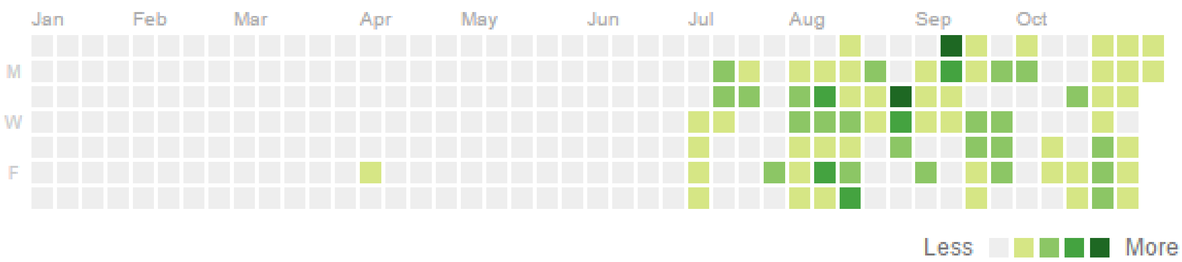


Figure 2: A screenshot of an interactive heatmap automatically generated by Github. Having the complete change history, consisting of 680 commits, Git can be used to provide all kind of metrics and visualizations of the development process. Shown here are my commits to the model repository. In May and June, I did not use Git yet and was also busy with research, exams, the CEF conference, and the cloud setup. One can also see a week of holidays in July, as well as a gap when focusing on the documentation in October. The single contribution in April was the upload of the old model of the Stolper-Samuelson effect in preparation for the conference.

Additionally, I believe that having better tools to provide and browse source code online would also help. The easier it is to gain insights by inspecting source code, the more researchers will do so.

In the case of using a professional version control system such as Git, there is also the nice side-effect of having a comprehensive lab journal, given the author commits and comments frequently. Furthermore, Git allows to tag these commits. This allows to name and refer to a specific version and configuration of the model. I make use of this capability by providing source tags in the form of #ExplorationChart to tag each discussed configuration. Given a tag, its source code can be simply browsed on Github by navigating to github.com/kronrod/agentecon/tree/ExplorationChart or the equivalent site for other tags. Anyone with the right tools installed can download and run that version and get the exact same result within minutes, which drastically improves replicability. Knowing such a tag also allows to interactively browse the outcome of the specified simulation by navigating for example to master.agentecon.com/sim.html?id=ExplorationChart (might or might not work depending on the browser).

2.4. Software Testing

> "The only way we validate a software design is by building it and testing it. There is no silver bullet, and no 'right way' to do design. Sometimes an hour, a day, or even a week spent thinking about a problem can make a big difference when the coding actually starts. Other times, 5 minutes of testing will reveal something you never would have thought about no matter how long you tried. We do the best we can under the circumstances, and then refine it."
>
> - Jack Reeves (2005)

Programmers make errors all the time. This can either be addressed preventatively by trying to write code with fewer errors, or by resolving them once they appear. Intuitively, one tends to put effort into avoiding errors in the first place. However, once the errors that can easily be addressed in advance are dealt with, it is more effective to accept ones imperfection and to focus on detecting and fixing errors as fast as possible. In combination with *fail fast* design as discussed by Shore (2004), this leads to a tight feedback loop between introducing and eliminating errors.

While programming, modern source code editors analyze and recompile the software all the time – allowing them to immediately mark syntax errors and other statically detectable problems on the fly, thereby enabling the programmer to fix them within seconds of having made them. As a next step, I use the tool Infinitest, which automatically identifies and runs all relevant tests on changes and marks faulty lines of code within seconds of saving the change. This allows to catch more sophisticated errors, such as violations of stock-flow consistency or deviations from expected equilibrium results.

In order to reasonable track down more subtle errors, the ability to exactly replay a simulation is essential. This reqires all random number generators to be deterministic and explicitely seeded (also one of the recommendations of Chang and Phillip (2015)). When striving for high replicability across different systems and over time, it is also advisible to choose a well-established programming language. Java, for example, has an excellent track-record of guaranteeing exact replicability across updates and across platforms, which is something that C++ for example does not. Rankings of programming languages such as that by Aruoba and Fernández-Villaverde (2014) often focus on performance only, neglecting replicability and other factors such as the ease of development and testing. The latter are much more valuable as they help saving expensive developer time as opposed to cheap computing time.

Once all local runs pass the tests, the changes can be committed to the code repository together with a suitable comment, thereby adding an entry to the lab journal. In professional environments, commits to the common repository often trigger further tests aimed at detecting whether the commit causes errors in combination with the latest commits of other developers, a practice called *continuous integration*. When shipping software to customer, the same tight feedback loop between detecting errors and fixing them is realized through the principle "release early, release often", which was introduced by Raymond (1999) in *The Cathedral and the Bazaar*.

2.5. Model Validation

Many authors choose a qualitative, empirical approach to verify their agent-based models. This is done by comparing the resulting output with well-known patterns. (Grimm et al., 2005) In economics, such patterns are usually referred to as stylized facts, an example being "firm sizes follow a Zipf distribution" (Axtell, 1999, 2001). Such criteria may hint at the qualitative plausibility of a model, but they do not provide much confidence in its accuracy. Often, the model is tuned until it subjectively looks right. Gatti et al. (2011) describe this process as follows: "The choice of parameter values has been constrained merely by the need to rule out patently unrealistic dynamic behavior, i.e. degenerating paths idenfiable by visual inspection and conventional empirical standards."

The vision of reproducing reality and thus also measuring the quality of the model by comparing its predictions to reality is understandable. Unfortunately, this approach can degenerate into a superficial validation of a few stylized facts. Given the high number of degrees of freedom of agent-based models, this is not satisfactory. Instead, quantitatively comparing agent-based models to equivalent equilibrium models provides a much more accurate and well-defined metric. To do so, a configuration is chosen that is computable under both paradigms. As long as an agent-based model is not shown to accurately replicate the well-known cases, the general confidence in its results remains muted.

Competitive validation is a promising option for more complex models that cannot be verified quantitatively. In competitive validation, different implementations of each agent type compete against each other in the same suite of simulations and are ranked according to a suitable metric. Competitive validation does not indicate when the optimum has been found, but it gives clear criteria to rank two proposed algorithms against each other.

2.6. Development Setup

From the beginning, the source code of the simulation has been separated from the source code of the running environment.[16] They are two different projects with an interface in between. This separation allows to update the visualization mechanism independently from the simulations. For example, when adding a new chart type on the website, this new chart automatically becomes available for all simulations, even those that have been written long before. The running environment consists of a backend that fetches tagged simulations from Github and runs them, and of a frontend that serves the website master.agentecon.com to the web browser. It runs on Google App Engine, to which it can be conveniently deployed with the click of a single button from Eclipse, the used code editor.

Ideally, this website would allow any reader of the thesis to browse the presented results in more detail. However, the large amounts of data bring the chosen tools to their limits, with App Engine being relatively slow at running the simulations and the highcharts library sometimes crashing when

[16] Hosted separately in a private bitbucket repository.

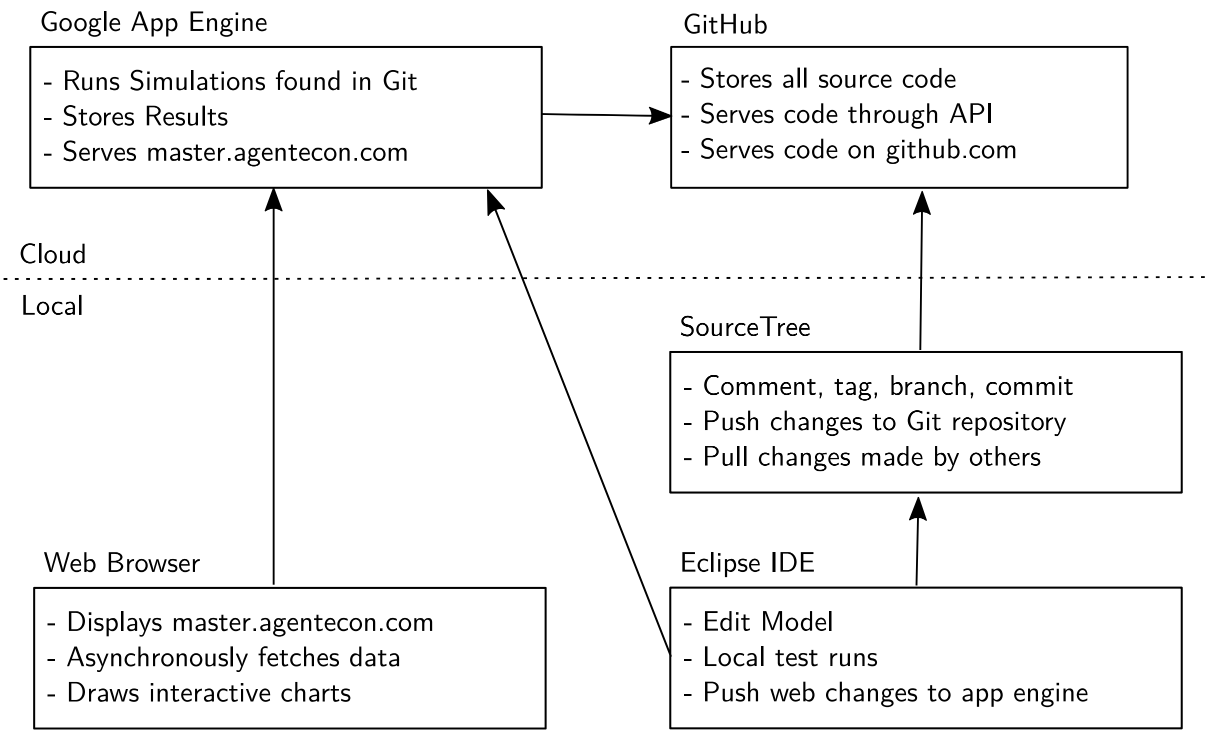


Figure 3: The development setup.

loading large charts. The interested reader can try out whether the website works reasonably in his browser by visiting master.agentecon.com/sim.html?id=ExplorationChart for browsing the outcome of the simulation #ExplorationChart or any other similarly tagged simulation by replacing the tag at the end of the url.[17] A printout of said website is attached as appendix A. Having interactively browsable charts proved of great value in developing and exploring the presented simulations.

The source code of the simulations is hosted on https://github.com/kronrod/agentecon, from where the simulation runner automatically fetches all tagged simulations and runs them as soon as they are requested for the first time. When working alone, a central code repository is a nice-to-have as it allowed me to work with the model even when not at home. In the case of multiple collaborating programmers, such a setup would be a must. Diagram 3 shows the discussed setup including Git client SourceTree mentioned in section 2.3. A more detailed description of the software architecture is attached as appendix B.

[17]Tested with Google Chrome and Microsoft Edge on a Windows 10 system.

3. Production Economy

This chapter describes an agent-based production economy, which serves as a foundation for the extensions discussed in the subsequent chapters. It consists of utility-maximizing consumers and profit-maximizing firms in a sequence of daily spot markets. Hens and Pilgrim (2002) motivate sequential markets as follows: "In no way can it be said that there is a complete system of contingent contracts that opens once for all times and that markets will remain closed ever after. Clearly, a more realistic setting is a model of sequential markets, i.e., a system of reopening spot markets [...]." As firms adapt their price beliefs over time and adjust daily production accordingly, the simulation converges towards pareto-efficiency and the same prices and volumes emerge as predicted by classic equilibrium theory.

The Stolper-Samuelson effect serves as a benchmark for the accuracy of the simulation. This effect requires a moderate level of complexity (multiple input and output goods) while still being simple enough to be numerically solved with standard methods. Section 3.1 describes the equation-based equilibrium model with two input factors and two output goods, with the subsequent sections presenting and discussing the agent-based version of the model.

3.1. Equilibrium Model

The Stolper-Samuelson theorem states that if the price of a good rises, then the input factor most intensively used in its production should rise along with it. (Stolper and Samuelson (1941)) Thus, we need an economy with at least two output goods and two inputs factors. For illustrative purposes, we call the two types of goods pizza and fondue. They are produced by according types of firms, pizzeria and chalet. There are also two types of inputs: Swiss and Italian man-hours, whereas Swiss man-hours are more intensively used for the production of fondue and Italian man-hours are more intensively used for the production of pizza. Both, the Swiss and the Italian consumers, have the same preferences. By default, they both prefer pizza. The consumers are endowed with 24 man-hours per day, part of which they sell on the market, buying pizza and fondue in return. Under these conditions, the Stolper-Samuelson theorem predicts that a rise of pizza prices will lead to a rise in Italian wages. Exogenous preference shocks are used to trigger such price shifts. As these shocks come unexpected to the agents and the economy is static otherwise, intertemptoral considerations are unnecessary and each configuration can be solved as an independent equilibrium in an Arrow-Debreu spot market.

Consumers derive utility from a log-utility function with consumed pizza, fondue and leisure as weighted inputs. The utility function of a single consumer of type $c \in \{Italian, Swiss\}$ is:

$$U_c(x_{c,pizza}, x_{c,fondue}, h_c) = \alpha ln(x_{c,pizza} + 1) + \beta ln(x_{c,fondue} + 1) + \gamma ln(24 - h_c + 1)$$

with α, β, and γ quantifying the preferences for each consumable, h_c denoting the man-hours sold on the labor market, and for example $x_{Italian,fondue}$ being the amount of fondue consumed by the Italian consumers. Note the increments $+1$ for each consumable to ensure that utility is always positive. Without them, a single consumer failing to acquire one of the inputs on a single day would suffice to drag the average experienced utility for all consumers down to $-\infty$, thereby spoiling average utility as a benchmark for the simulation.

Consumers maximize utility subject to their budget constraint

$$w_c h_c + d = p_{pizza} x_{c,pizza} + p_{fondue} x_{c,fondue} \tag{1}$$

with w_c denoting wage per hour for consumer type c, d being dividends per consumer, and p standing for price.

Given prices, firms produce to maximize profits, which they distribute evenly to the consumers as dividend. They have a Cobb-Douglas production function with decreasing returns to scale in order to rule out monopolistic equilibria. Pizzerias (*piz*) have production function 2, chalets (*cha*) have production function 3.

$$x_{pizza}(h_{Italian,piz}, h_{Swiss,piz}) = A\ h_{Italian,piz}^{\lambda_{high}} h_{Swiss,piz}^{\lambda_{low}} \tag{2}$$

$$x_{fondue}(h_{Italian,cha}, h_{Swiss,cha}) = A\ h_{Italian,cha}^{\lambda_{low}} h_{Swiss,cha}^{\lambda_{high}} \tag{3}$$

Parameters A, λ_{low}, and λ_{high} are constant, with $\lambda_{low} < \lambda_{high}$, and $\lambda_{low} + \lambda_{high} < 1.0$. The profit function of a pizzeria is provided later as equation 4.

In the agent-based simulation, each of the hundreds of consumers and firms acts on its own. In the general equilibrium case, each agent type is represented by one representative agent whose consumed and produced quantities are scaled to the actual number of agents.

A script to calculate the general equilibrium solution is provided as supplement (see section **??**).

3.2. Agent-Based Model

Instead of explicitely imposing equilibrium conditions, agent-based models delegate that work to market forces, hoping for equilibria to emerge naturally.

The simulation approaches its equilibrium over the course of many iterations (days), forming a sequence of reopening spot markets with nightly production. Money is introduced as a store of value and to facilitate trading. Consumer and firm agents have the same utility and production functions as in the general equilibrium model. Preferences are set per consumer type and production parameters per firm type. However, each individual agent has its own stocks of money, pizza, fondue and man-hours, whereas Italian and Swiss man-hours are traded as distinctive goods. Furthermore, each firm has its own price beliefs at which it posts offers (bids and asks) to the market. The source tag of this configuration is #ComputationalEconomicsPaper.

3.2.1. Money

While absent in the equilibrium model, which is only concerned with relative prices, money serves an essential purpose in the agent-based model. Money reduces the computational complexity of finding pareto-improving trades in an Arrow-Debreu economy, as Feldman (1973) found out (although without using the term *computational complexity* yet). He proved that as long there is a good that every agent owns and values – namely money –, bilateral trading suffices to reach an efficient equilibrium. Without money, it might be necessary to identify mutually beneficial trades involving more than two parties, for example if ten agents in a circle each own what the agent to their right desires.[18]

Note that technically, the money in the presented model does not fulfill Feldman's criteria of money, as it is neither directly valued by the consumers, nor do all consumers hold money all the time. However, this does not prevent the simulation from reaching the efficient outcome. Firstly, even though money does not enter the consumers' utility functions, they can value it indirectly by seeing what consumption goods money could buy them on the market. Furthermore, as long as they can obtain money by working at any time, they can behave as if they already held that money. Therefore, Feldman's results are still be applicable. In short: thanks to money serving as a transitory store of value, hard-to-find instantaneous k-lateral trades can be safely split into a sequence of easy-to-find bilateral trades.

[18] With n agents, there are only $n(n-1)/2 = O(n^2)$ possible bilateral pairings, but $2^n - n - 1 = O(2^n)$ potential multilateral trades. Thus, having money reduces the problem of finding a valid trade from exponential to polynomial complexity.

3.2.2. Sequence of Events

Like the real world, agent-based models do not permit instant market clearing. Instead, trades and other events happen in chronological order. Circular dependencies are broken apart. For example, firms cannot sell output goods they have not produced yet, requiring them to sell yesterday's production today and today's production tomorrow.

Furthermore, the dynamics of a simulation are affected by causality, which is irrelevant to the equilibrium solution. For example, the equation $d = \pi$ does not distinguish cause and effect. But in the simulation, it makes a difference whether dividends determine profits or profits determine dividends. Here, the sequence of events plays a pivotal role.

Each firm is endowed with 1000$ before the first day begins. Then, days are structured as follows:

1. Consumers are endowed with 24 man-hours each.
2. Firms distribute excess cash dividends. As a first step, excess cash is defined as all cash above a given threshold τ. More elaborate heuristics are discussed in section 4.3.
3. Firms post asks to the market, offering yesterday's production in accordance with their individual price beliefs; for example *"we sell 79 pizzas for 7.30$ each"*.
4. Given their price beliefs and available cash, firms choose a production target and accordingly post bids in the form of limit-orders to the market, for example *"we buy up to 50 Swiss man-hours for 13$ each"*. By default, the chosen production target maximizes profits.
5. In random order, consumers enter the market and optimize their utility given the offers they find, selling man-hours and buying pizza and fondue.
6. The market closes and each firm updates its price beliefs based on whether the relevant orders were filled or not.
7. Firms use all acquired man-hours to produce the outputs to be sold tomorrow. Unsold outputs are carried over to the next day, whereas unused man-hours cannot be stored. In equilibrium, all money resides with the firms again at this point in time, although not necessarily equally distributed.

An alternative pricing mechanism is markup-pricing and is for example optionally available in the crisis model.[19] With markup-pricing, firms have a belief regarding the optimal production and set prices at a markup above production costs. However, given the assumption of firms being price-takers (i.e. Bertrand competition), basing the firms' volume decisions on price-beliefs is a more natural choice. Markup-pricing is not further discussed in this thesis.

3.3. Exponential Price Search

In agent-based simulations with endogenous price-discovery, firms typically have price-beliefs that are updated heuristically, based on whether a market offer based on the current price belief was filled. For example, a pizzeria that offered 200 pizzas for 11$ each will adjust the price upwards if it succeeds in selling them and will adjust the price downwards if not. The case of a partially sold inventory can be neglected as this only happens rarely with large enough numbers of competing firms, an observation already made by Gintis (2007).

[19] github.com/crisis-economics/CRISIS/blob/master/CRISIS/src/eu/crisis_economics/abm/firm/MacroFirm.java

3.3.1. Conventional Methods

Many simulations adjust prices by a certain percentage, i.e. $p_{t+1} = (1 \pm \delta)p_t$, examples being Gintis (2007), Catalano and Di Guilmi (2015) and Gatti et al. (2011). One shortcoming of this approach is that it is not symmetric. Increasing a price p and decreasing it again counter-intuitively leads to $p_{t+1} = p_t(1+\delta)(1-\delta) \neq p_t$.

To ensure symmetry, one should multiply or divide by a constant factor instead, i.e. $p_{t+1} = p_t(1+\delta)$ or $p_{t+1} = p_t/(1+\delta)$. This is one of the available pricing options in the crisis-economics model.[20] However, this method can still suffer from coarse granularity and a biased average. To see this, first note that in a situation with stable prices, symmetry dictates beliefs to be too high and too low equally often. If this was not the case, price beliefs would move over time, contradicting the assumption of stable prices. As an example, assume a market price of 101$ and an adjustment factor of $(1+\delta) = 1.05$. Starting with 100$, a firm's price belief will alternate between 100$ and 105$. The average price belief among the firms will thus be around 102.5$, above the market price of 101$. Riccetti et al. (2014) overcome this bias through randomization. For example, a new δ_{rand} could be chosen uniformly for every step, with $\delta_{rand} \sim U(0, 2\delta)$.[21]

When chosing the adjustment factor, there is a trade-off between speed of convergence and accuracy. A large factor lets the price belief approach the market price faster, while a small factor allows for higher accuracy. Generally, the number of steps it takes to converge is linear in the relative logarithmic distance between price belief and market price, i.e. it is in $O(|log_f(p_{belief}/p_{market})|)$. In practice, it can also take even longer depending on the competitive dynamics between firms.

3.3.2. Exponential Search

Exponential search is an algorithm that efficiently solves the unbounded search problem by dynamically adjusting its step size. It finds a target element in an unbounded list in logarithmic time, i.e. in $O(log(d))$ with d being the number of steps it would take with a linear search. Exponential search was first described by Bentley and Yao (1976) and is well-known among computer scientists. Since the firm's problem of finding a market price is also a search in an unbounded, one-dimensional space, employing exponential search is a natural choice. Wolffgang (2015) seems the first to do so in agent-based economics.

Classic exponential search doubles the adjustment factor on every step until it passes by the target value and then switches into bisection mode. To allow for dynamics, Wolffgang (2015) suggests to generally increase the adjustment factor on steps in the same direction as before and to decrease it on turns. Unfortunately, this can lead to cycles, thereby preventing convergence, as shown in figure 5. We address this by only doubling after every second step in the same direction, leading to the algorithm illustrated in figure 4. Furthermore, doubling and halving might be too aggressive, potentially causing or amplifying oscillations. Wolffgang applies a factor of 1.1, a value which we adopt.

3.4. Sensor Prices

While exponential search helps to achieve faster convergence and better accuracy, it does not address input synchronization. Input synchronization being hard to achieve in price-driven markets has also

[20] See class `ReducePriceIfUnsoldPricingAlgorithm` in their repository github.com/crisis-economics

[21] To be precise, Riccetti et al. randomize the percentage approach, i.e. $p_{t+1} = (1 \pm \delta)p_t$ with δ uniformely distributed, leading to a forth variant not discussed here.

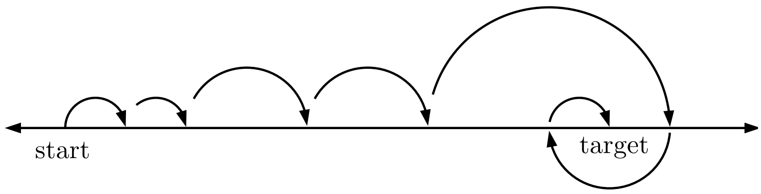


Figure 4: Adapting price beliefs with exponential search: increasing the adjustment factor after every second step in same direction, decreasing on turns

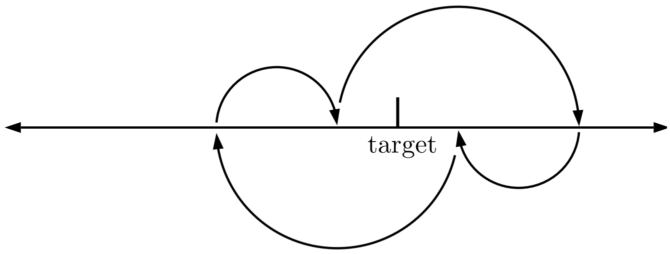


Figure 5: Trap: no convergence when increasing the adjustment factor too early

been observed by firm theorists Milgrom and Roberts (1994). For firms depending on multiple perishable input goods (here different types of man-hours), it is essential that all their bids succeed. With Cobb-Douglas production, failing to acquire one of the input goods already leads to a total loss of production. Sensor prices improve input synchronization.

In equilibrium, half of the offers will not fill when adjusting symmetrically up- and downwards as described in the previous section. In particular, firms with multiple, perishable input factors suffer from poor input synchronization, a phenomenon described by Milgrom and Roberts (1994). A pizzeria with Cobb-Douglas production that fails to either acquire Swiss or Italian man-hours, will produce nothing at all on that day. Coase (1937) suggests to address market frictions by introducing long-term contracts, which is also the preferred solution in reality when acquiring man-hours. Customer loyalty can also help, as Rouchier (2013) demonstrates with an agent-based simulation. To preserve the elegance of a sequence of independent spot markets, we decided to apply a new method, which we call *sensor prices*.

Normally, when posting an order to the market, agents face a trade-off between information exploitation and information exploration, as Tesfatsion (2006) points out. In the case of a sale, they want to maximize revenue, but also collect as much information as possible about the optimal price level. These two conflicting goals can be disentangled by posting two seperate offers, one that maximizes revenue and one to find out what prices the market can bear.

Figure 6 illustrates how only every second order is filled when using typical price adaption heuristics. Figure 7 shows how sensor prices can improve the situation. The sensor offer constantly tests the price level and adjusts itself accordingly. It uses a fraction θ_s of the total sales volumes, whereas the majority of the output is sold at a close, yet safe, relative distance θ_d, leading to prices $p_{volume} = p_{sensor}/(1+\theta_d)$ when selling and $p_{volume} = p_{sensor}(1+\theta_d)$ when buying. For simplicity, we impose $\theta_s = \theta_d = \theta$.

In order to find the right distance between sensor price and volume price, their relative distance θ_d is dynamically adapted. Whenever the volume offer fills, it is cautiously moved a little closer to the sensor offer, with $\theta_{t+1} = \theta_t/1.005$. However, if it does not fill, distance is doubled to keep the risk of repeated failures low, i.e. $\theta_{t+1} = 2\theta_t$. With this strategy, one can expect the ratio of failures to be $1/log_{1.005}(2) < 1\%$. These parameters have been set intuitively by trial and error, without further

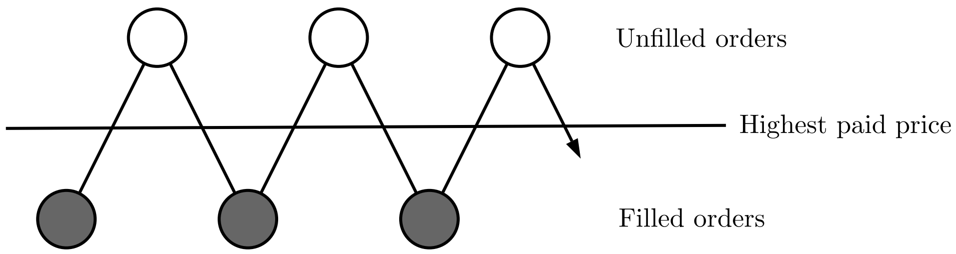


Figure 6: Typical price adaption heuristics lead to filled orders only half of the time, alternating between a price below and above what the market can bear.

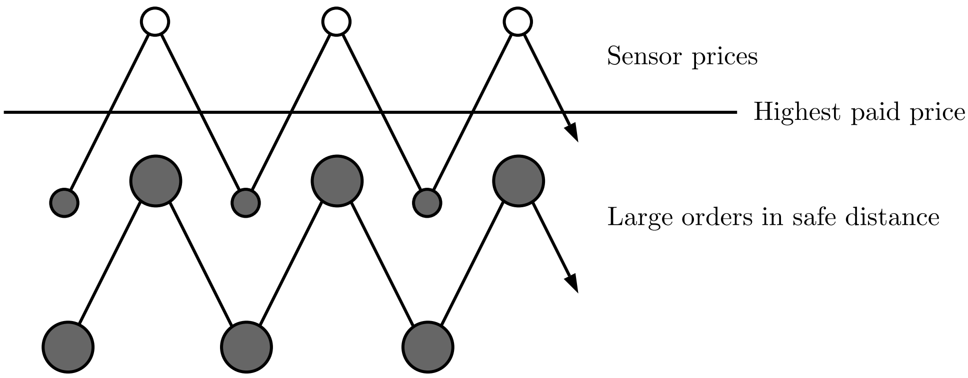


Figure 7: With sensor prices, only a small fraction of volume is sacrificied for price exploration, whereas the bulk can reliably drive revenue.

analytic evaluation.

In management science, sensor prices would likely be considered a form of dynamic pricing, as for example researched by Elmaghraby and Keskinocak (2003). However, they differ insofar as dynamic pricing is primarily concerned with price discrimination, which is the art of selling the same product at different prices depending on the consumer, while sensor prices are tailored towards information exploration in an open market with indistinguishable consumers.

3.5. Results

The agent-based simulation exhibits the Stolper-Samuelson effect as an emergent property with high accuracy. Within the parameter space of $\alpha \in (1.0, 9.0)$, simulated relative prices deviate by 0.02% from the general equilibrium benchmark on average.[22] This high level of accuracy is only achieved when employing all the three discussed techniques in combination.

The default configuration assigns the parameter values shown in table 1. No exogenous shocks are included as long as we are only concerned with the asymptotic outcome, which does not depend on the point in time the relevant parameter values are set.

[22]Wages deviated a little more, namely by 0.03% on average, as discussed later. The most extreme outlier was observed for $\alpha = 1.2$ with an error of 0.48% for relative output prices.

Symbol	Value	Description
n_c	100	Consumers of each type
n_f	10	Firms of each type
α	7	Consumer preference for pizza
β	$10 - \alpha$	Consumer preference for fondue
γ	14	Consumer preference for leisure
λ_{high}	0.375	Primary input weight
λ_{low}	0.125	Secondary input weight
$\lambda = \lambda_{high} + \lambda_{low}$	0.5	Labor share of income
A	10	Productivity
$c_{f,1}$	1000	Initial cash holdings of each firm f
τ	800	Excess cash threshold
$d = n_f(c_{f,1} - \tau)/n_c$	20	Resulting dividends per consumer
δ	0.03	Step size of price adaption
	[0.001, 0.5]	Bounds of δ with exponential search
	1.1	Adaption factor for δ in exp. search
θ	[0.001, 0.5]	Bounds of sensor distance and volume
	1.005	Divisor for gradually decreasing θ
	2	Factor for increasing θ
	2000	Number of simulated days
	[1001,2000]	Relevant time span for benchmark

Table 1: Parameters

3.5.1. Algorithm Comparison

Table 2 compares the accuracy exponential search to those of conventional adaption algorithms, with exponential search being the clear winner. Generally, prices of goods tend to be more accurate than wages and trading volume tends to be the least accurate. The measured prices are volume-weighted averages. This increases accuracy a little as mispricings normally also come with reduced trading volumes, and thus have a lower weight in the metric. Surprisingly, increasing the number of agents per type does not necessarily lead to more accurate results. Intuitively, one would expect consistently higher accuracy with larger populations due to the law of large numbers. Investigating the driving forces behind these differences might be a topic for future research.

Method	$\frac{p_{pizza}}{p_{fondue}}$	Error	$\frac{p_{pizza}}{w_{Swiss}}$	Error	x_{pizza}	Error
Constant percentage	1.713543	0.116%	2.676847	0.765%	611.956	4.279%
Constant factor	1.713371	0.126%	2.678184	0.813%	607.736	4.939%
Randomized factor	1.711958	0.208%	2.677775	0.798%	633.174	0.960%
Exponential search	1.715483	0.003%	2.657247	0.025%	639.168	0.022%
Benchmark	1.715526		2.656574		639.311	

Table 2: Accuracy of price adaption methods in a typical scenario. Relative prices of consumption goods tend to be more accurate than those of wages. Errors relative deviations from the benchmark.

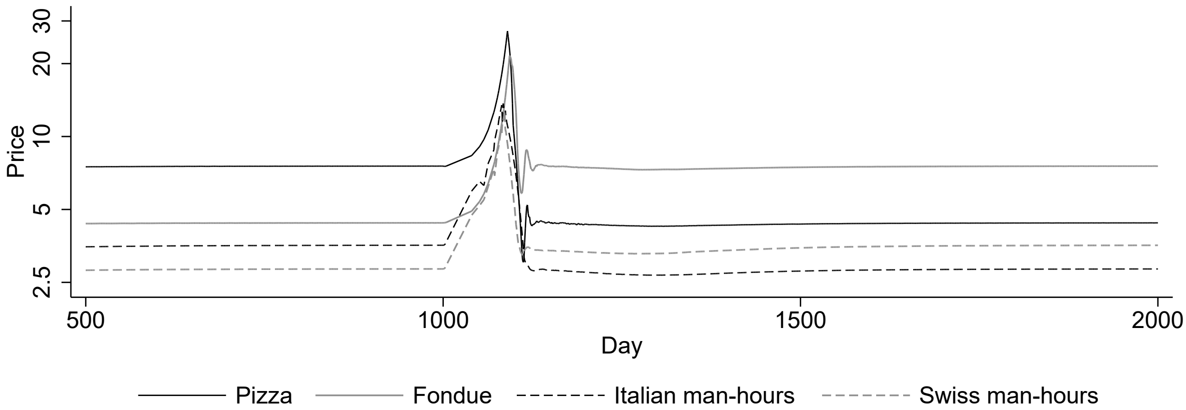


Figure 8: Price dynamics with enabled sensor prices, exponential search, and dividend-based normalization. It takes a little more than 100 days to find the new equilibrium after an exogenous preference shock on day 1001.

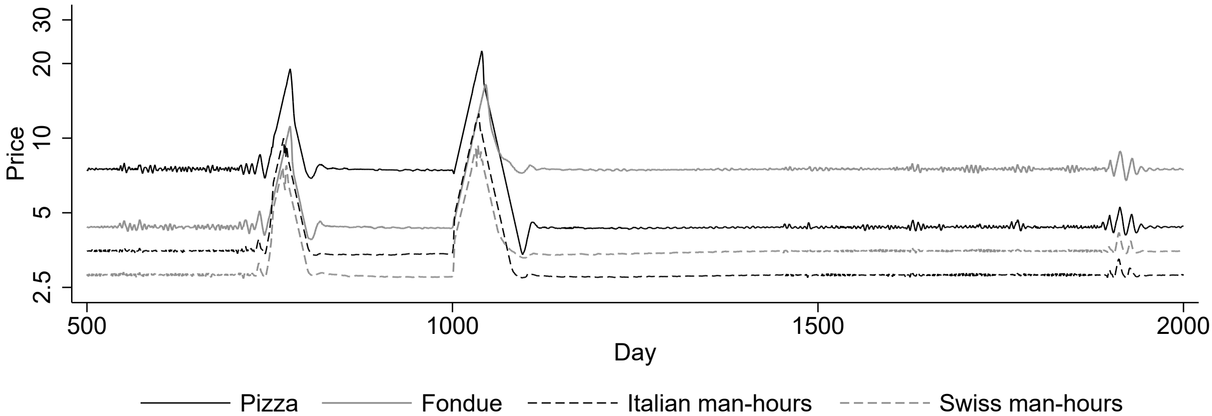


Figure 9: Switching from exponential search to constant percentage adjustment, accuracy and stability are reduced.The escalation after day 1000 is caused by the exogenous preference shock, the others are triggered by small perturbations due to the randomized order in which consumers enter the market each day.

3.5.2. Preference Shocks

To investigate the dynamic behavior of the simulation, a preference shock is introduced on day 1001. That day, consumers wake up suddenly preferring fondue over pizza, with swapped preference parameters α and β. Prices after the shock approach the same values as those before, except that the new pizza price is the old fondue price and vice versa. Furthermore, as the Solper-Samuelson effect predicts, Italian and Swiss wages also switch. Figure 8 shows prices over time in the default configuration. It is accurate and stable, although there is a period of turmoil after the preference shock, during which production breaks down. Without production, there is not much to buy and thus no incentive to work either - contributing further to the decline of production. At the same time, consumer wallets are still refilled daily by dividend payments, leading to escalating prices until work pays off again, production recovers, and prices rebalance. Switching from exponential search to any of the other three adjustment methods, accuracy is reduced and sporadic deviations start to occur endogenously, as shown in figure 9 with constant percentage adaption. Due to the random queueing of the consumers, there are constant

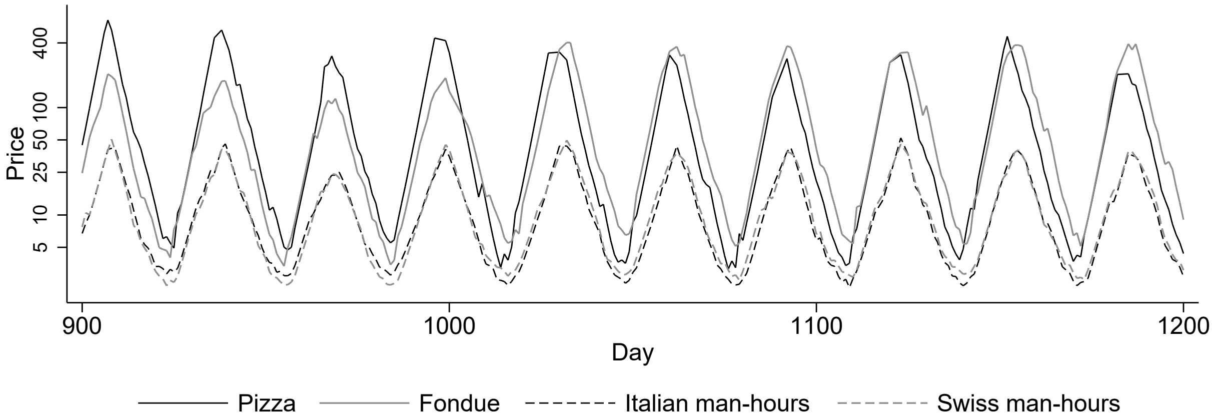


Figure 10: In the default configuration, disabling sensor pricing leads to perpetual oscillations and a significantly weakened Stolper-Samuelson effect.

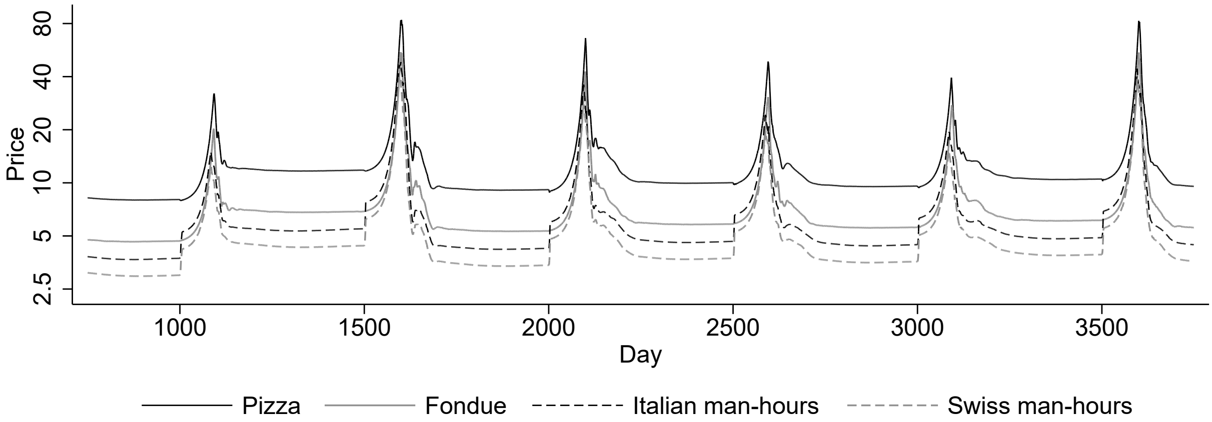


Figure 11: Without price normalization, absolute price-levels can change after shocks. The shocks are exogenously triggered by temporary preference changes. With normalization enabled, prices always return to the same nominal level.

small perturbations that can trigger endogenous price escalations. In contrast, exponential search is not as easily thrown off balance. While the Stolper-Samuelson effect can be observed well with all four adjustment strategies, it is less apparent when disabling sensor prices as shown in figure 10. In contrast, the simulation can still stabilize without price normalization, although not on a predictable price level as shown in figure 11.

3.5.3. Computational Complexity

Agent-based simulations can be seen as just another numerical method of finding equilibria. In this context, it is of interest to compare its performance to other numerical methods. While no method can perform better than the theoretic lower bound, there can still be enormous differences between different methods in practice. Furthermore, under special conditions, the computational complexity of solving a specific problem can be dramatically lower than the generally valid lower bound suggests. I find that the agent-based simulation is much faster than numerically solving the equivalent equation-based model by constrained optimization.

Etessami and Yannakakis (2010) introduce the FIXP complexity class and show that computing a fixed point of a Brouwer function is in that class, i.e. FIXP-hard. In their seminal paper, Arrow and Debreu (1954) showed that finding equilibrium prices is such a fixed point problem. Thus, finding equilibrium prices is FIXP-hard in the general case. FIXP is related to the better known complexity class NP as it also takes non-polynomial time to calculate the solution, but differs insofar as a solution is known to exist from the beginning. Simply put, this means that the classic Walrasian auctioneer has a difficult job. Finding the equilibrium prices in a scenario with thousands of commodities and agents is infeasible with modern computers in general.

Axtell (1999) sees this high computational cost as a hint that the centralized model of the Walrasian acutioneer is not realistic. Instead he proposes a decentralized algorithm of finding a pareto-efficient equilibrium much faster, namely in polynomial time. This is achieved by restricting exchange to k-lateral trades and assuming they these trades can be found in constant time. This might be more realistic, but one is not guaranteed any more to find the efficient equilibrium. This sidestepping of the computationally hard part was also observed by Ghosal and Porter (2013).

Fortunately, the problem is much simpler in markets with money - whereas money is defined as a good that every agent owns. Feldman (1973) showed that bilateral trades suffice to find a pareto-optimal equilibrium as long as every agent still holds a positive amount of money in the end, and as long as every agent values money. Similarily, bilateral trade also suffices if there is at least one agent who owns a positive amount of every good. (Rader, 1968) This insight was later refined by Goldman and Starr (1982), showing that a pareto-optimum can only be reached with bilateral trades if there is a sufficiently large overlap in the agents' inventories.

Even though the tested model does not perfectly fulfill Feldmann's criteria of money as consumers do not derive utility from it directly, it is still probable that the simulation's efficient equilibrium is much easier to find than in the general case. A strong hint for that is the fact that the time the simulation takes to find the equilibrium does not grow measurably as the complexity of the model is increased, as shown in table 3. At the same time, numerically solving the equivalent equation-based model through constrained optimization takes exponentially longer as the number of consumer and firm types is increased.[23]

These results indicate that agent-based models can be a competitive option at finding equilibria - at least when benchmarked against the standard approach. Future work could further benchmark the agent-based simulation against optimized algorithms such as Negishi's method or the ones described by Scarf and Hansen (1973). Generally, if there exists a fast way to find a solution through an agent-based simulation, there should also be a similarly fast numerical method as both are bound by the same theoretical limits.

[23]To find the numerical solution, the JaCoP solver[24] is fed with the analytically derived equilibrium conditions.

Model		Constrained Optimization		Agent-Based Simulation		
n_f	n_c	Variables	Time	Time	Error	Source Tag
1	1	32	0.12 s	0.55 s	0.14%	#Benchmark11
1	2	62	0.84 s	0.50 s	0.32%	#Benchmark12
1	3	92	1.7 s	0.57 s	0.61%	#Benchmark13
2	1	54	0.21 s	0.54 s	0.18%	#Benchmark21
2	2	104	3.2 s	0.6 s	0.37%	#Benchmark22
2	3	154	54 s	0.88 s	0.10%	#Benchmark23
3	1	76	0.31 s	1.5 s	0.35%	#Benchmark31
3	2	146	28 s	0.95 s	0.17%	#Benchmark32
3	3	216	3035 s	0.98 s	0.35%	#Benchmark33

Table 3: As complexity increases, the time it takes to numerically solve the equation-based model with general methods grows much faster than the time it takes for the agent-based simulation to converge towards the same solution.

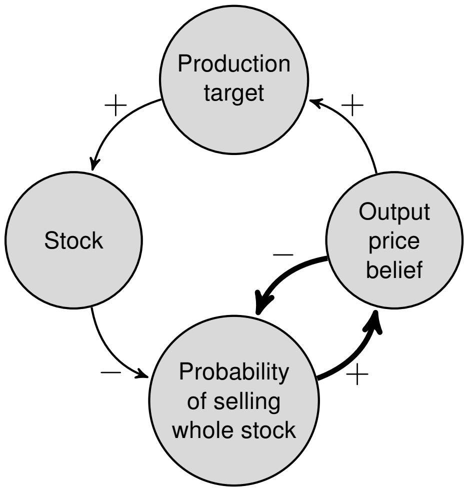


Figure 12: The causal loop diagram for a firm's output price belief has two balancing feedback loops.

4. Model Dynamics

In order to reach stability, the agents' decision rules must not only work in equilibrium situations, but also when the simulation is out of equilibrium. Well-designed decision rules contribute to the overall stability of the model, whereas less well-designed decision rules can cause instability – even though they both might be identical in equilibrium. This chapter discusses how the firms' decision rules affect the system's dynamics, in particular its pricing heuristics as well as its dividend decisions.

4.1. Pricing Dynamics

One method of classifying models as stable or unstable is to calculate their Lyapunov exponent, as mentioned by Axtell (2005) and described by Hommes (2013). Herein, the much simpler causal loop diagrams suffice, which we use in accordance with the guidelines of Kim (1992). Causal loop diagrams allow to quickly reach a qualitative judgement on whether a feedback loop is reinforcing (unstable) or balancing (stable). Undesired reinforcing feedback loops are colloquially called *vicious cycles*. Causal loop diagrams visualize system variables as nodes in a directed graph. Edges are either labeled with a $+$ or $-$, depending on whether an increase of the originating variable leads to an increase or decrease of the target variable. In such graphs, feedback loops passing an even number of minusses are reinforcing, while those with an odd number are balancing.

Figure 12 shows the causal loop diagram for a firm's price belief regarding the output good. A firm that believes it can sell at a higher price will try to produce more, thus increasing its production target. A higher production target subsequently results in a higher actual production and a larger stock of goods to be sold. However, the higher the stock, the less likely it becomes to fully sell it on the market. Additionally, trying to sell the stock at a higher price also leads to a reduced sale probability. Applying one of the belief adjustment heuristics discussed in section 3.3, a high sale probability results in a higher price belief, thereby closing the two loops. Note that both loops are balancing, and thus stabilizing the system.

The price dynamics for the input good are similar and illustrated in figure 13. Here, a low price belief leads to an increased production target. A firm should produce more as its input factors are getting

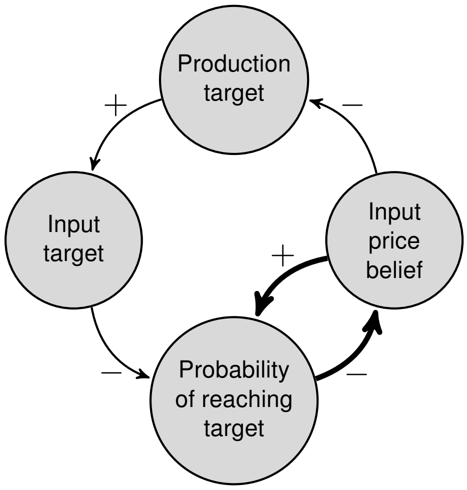


Figure 13: Causal loop diagram for input price beliefs, also containing two balancing feedback loops.

cheaper. The higher production target calls for acquiring larger input quantities, which in turn makes a successful purchase of that increased input amount less likely. A decrease in that probability pushes the price belief upwards via the algorithms specified in section 3.3, thereby closing the outer loop. The inner feedback loop connects the input belief directly with the probability of reaching the purchase target, as offering a higher price makes it more likely that enough willing workers are found.

Thus, both, the feedback loop around the input factors as well as that around the output goods, contribute to stabilizing the system's dynamics.

4.2. Price Normalization

While money is a necessity in agent-based models (see section 3.2.1), only relative prices are usually considered in equilibrium models. There, goods and services are produced, traded, and consumed instantaneously, making money unnecessary. Even when explicitely introducing money, Sims (1994) shows that absolute price levels stay indeterminate - regardless of money supply. In order to resolve the indeterminacy of prices, at least one price must be set endogenously. In equilibrium models, this is usually done by normalizing the price of a randomly chosen good to one.

Doing the same in an agent-based simulation is not advisable as it can reduce the stability of the system. For example, imposing $p_{pizza} = 10\$$ would interrupt that price's two balancing feedback loops, thereby leaving all the work of approaching the equilibrium to the input side and to the other firm types. This is analogous to a central bank trying to control price levels by setting the price of bread to one and waiting for all other prices to adjust accordingly.

While it is possible to not normalize prices at all and let the simulation settle on a random price level as shown in figure 11, there is an alternative way of price normalization that comes with the benefit of additionally stabilizing the system. Instead of basing dividends on profits, we let firms distribute all their cash holdings above a given threshold as dividends. Observing that all money resides with the firms at the end of each day, and setting the threshold low enough, this policy effectively makes daily dividends a constant.[25] Besides binding nominal prices to money supply, this policy also improves

[25] With threshold τ, total dividends of n firms f with cash c_f each are $d_{tot} = \sum_f (c_f - \tau) = \sum_f c_f - n\tau$, which is money supply minus another constant term.

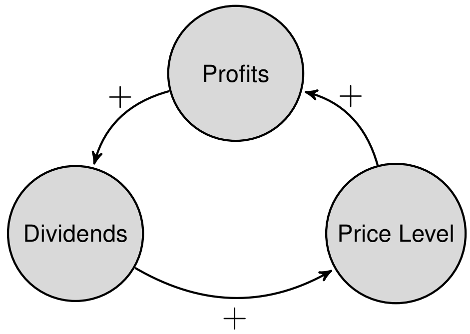


Figure 14: A vicious cycle with self-reinforcing inflation or deflation.

stability by breaking the vicious cycle of dividends, profits, and prices from figure 14.

$$\pi_{piz} = R_{piz} - C_{piz} = p_{pizza} x_{pizza} - \sum_c w_c h_{c,piz} \tag{4}$$

This vicious cycle can also be understood analytically. First, nominal profits rise with the price level, as can be intuitively seen from the profit function 4 of the pizzeria. When prices and wages are increased by a constant factor, equilibrium profits Π_{piz} grow by the same factor. Second, dividends are usually set equal to profits, extending the proportional dependency to dividends. Third, the vicious cycle is closed by recognizing from the consumer's budget constraint (equation 1) that equilibrium prices are proportional to the consumer's cash holdings at the beginning of the day,[26] which happens to consist entirely of dividends.

4.3. Seemingly Equivalent Heuristics

This section discusses dividend calculation heuristics that are equivalent in equilibrium, but that have different dynamic properties.

4.3.1. Setting

Given a trivial Cobb-Douglas production function $x(h) = Ah^\lambda$ without capital, profit maximization results in a labor share λ and a profit share $1 - \lambda$. In case of multiple input goods, the labor share can still be denoted as $\lambda = \sum_c \lambda_c$ with λ_c being the elasticity of input c. Both profits π and costs C can be expressed as a fraction of revenue, with $\pi = (1 - \lambda)R$ and $C = \lambda R$, together implying $\pi = \frac{1-\lambda}{\lambda}C$. Together with profit function 4, this results in three ways to calculate profits:

$$\pi = R - C = (1 - \lambda)R = \frac{1 - \lambda}{\lambda}C$$

Linearly combining them, any choice of finite coefficients a_π, a_R in equation 5 must lead to the same optimal result in equilibrium.

$$\pi = a_\pi(R - C) + a_R(1 - \lambda)R + (1 - a_\pi - a_R)\frac{1 - \lambda}{\lambda}C \tag{5}$$

[26]A quick path to this insight is to imagine the consumer being endowed with one gold nugget of market price d instead of dividends d. This yields the same outcome, yet transforms d into a price that must be – like every price – proportional to the general price level.

This equation is somewhat redundant and can be transformed into a the trivial linear combination

$$d = b_R R + b_C C \tag{6}$$

whose coefficients b_R and b_C have to fulfill equation 7:

$$\lambda(b_C + 1) = 1 - b_R \tag{7}$$

While in equilibrium, any choice of b_R yields the same result, the agent-based model usually is slightly off the equilibrium, leading to different dynamics depending on the choice of b_R. For some values, it will not converge at all.

To complicate matters further, it is not obvious how a firm should measure R and C. At the point in time at which the dividends d_t are determined, no trade has taken place yet and R_t as well as C_t are not known yet. Instead, the decision could for example be based on yesterday's R_{t-1} and C_{t-1}, or the firm could calculate the expected revenue $E[R_t]$ given its price beliefs and stock. As costs, the planned spendings $E[C_t]$ could for example be plugged in. There is a multitude of additional thinkable options, resulting in a zoo of of slightly different dividend heuristics.

4.3.2. Benchmark Scenario

A benchmark scenario #ExplorationChart is constructed in order to evaluate the different dividend heuristics. It is based on the production economy from section 3, having the same two consumer and firm types. Firms are given a higher labor share of $\lambda = 0.7$ in order to break symmetry. Furthermore, firm spendings are set to a constant fraction $f_s = 0.2$ of cash holdings, with dividends being subject to the tested heuristics instead. Binding spendings to money supply again determines the nominal price level. As the labor share is proportional to the profit share in equilibrium, nominal prices are again directly proportional to total money supply, as well as to the spending fraction f_s.

As a metric for efficiency, average experienced daily utility is chosen for its universality. An alternative metric would be real firm profits. Both metrics raise the question of how to aggregate across types. Pareto-efficiency can coincide with maximum average utility, but it generally does not, as it only guarantees that utility cannot be raised further through mutually beneficial trades. Thus, a heuristic that scores a better average utility is not necessarily better at optimizing each firms profits. However, the best tested heuristics approximately reach pareto-efficiency, allowing to ignore this concern here.

The benchmark scenario runs for 2000 days, with the initial 250 days being discarded for the utility measurements, giving the simulation some time to find its initial equilibrium. There are four small preference shocks of increasing magnitude at day 1000, 1250, 1500, and 1750. They simultaneously increase the fondue preference and decrease pizza preference for all consumers by 0.7 in total, or 0.1, 0.15, 0.2, and 0.25 in each step. These shocks are big enough to derail the less stable heuristics, but small enough to be handled without productivity crisis in the case of the more stable heuristics.

4.3.3. Evaluation

This section uses the previously defined benchmark to discuss the characteritics of three exemplary heuristics chosen for their simplicity and good performance. Figure 15 shows the average utility that was achieved with each of the heuristics as b_R was varied. The stated observations have been verified with other parameter values of $\lambda \in [0.05, 0.95]$.

1. The '*known*' heuristics uses yesterday's revenue R_{t-1} and costs C_{t-1} as they are the most recent known values. It performs well early on, but quickly detoriates for $b_R \geq 1.0$, a value which corresponds to the standard profit function $d = R - C$. Thus, it is not advisible to use the standard profit

function in combination with the latest observed values. Better and more stable results are achieved with $b_R < 1.0$. Its peak utility levels are visibly below the optimum.

2. The '*optimal cost*' heuristics has the opposite behavior, failing for low values b_R and striving on large ones. It fails to produce meaningful results for $b_R < 1 - \lambda$, where b_C is positive. This heuristic uses a revenue estimate $E[R_t]$ based on price belief and amount of goods to be sold, and the level of spending $C_{t,opt}$ that would maximize profits given the current price beliefs. Considering that $C_{t,opt} > C_t$ holds when the firm has too little cash and $C_{t,opt} < C_t$ holds when the firm has too much cash, its behavior is not surprising: A positive b_C leads to generous dividends exactly when there is not enough cash, therefore pushing the firm away from equilibrium. The opposite is the case for negative b_C, which lets the firm restrict the dividend when its cash levels are low so it can move towards the efficient equilibrium.
3. The '*expected*' heuristic is also based on $E[R_t]$, but uses planned spendings $E[C_t]$. Its economy never breaks down completely for all tested $b_R \in [-5, 5]$, and chosing $d = (1 - \lambda)R$ seems to score well also for other λ. For this specific parameter choice, the 'expected' heuristic is identical to the 'optimal cost' heuristic as $b_C = 0$ and costs do not enter the equation.

Further tests included sliding averages of past values, optimal dividends given prices, as well as expected revenues given planned costs. In particular, paying dividends that correspond to the efficient equilibrium given price beliefs fails miserably. All in all, a good choice seems:

$$d = (1 - \lambda)E[R_t] \tag{8}$$

It is simple, more stable than the standard profit equation, and performs reasonably well under various conditions. Although sometimes performance can be improved by considering this rule a special case of the 'optimal cost' heuristic and tuning b_R, my recommendation is to stick with the simple equation.

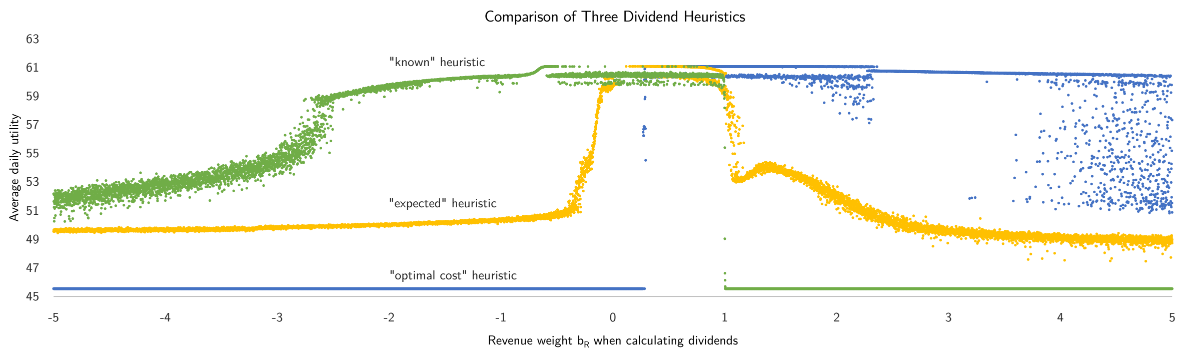


Figure 15: Each dot represents one of 30'000 simulation runs. Three different variants of measuring revenue R and cost C in order to calculate a firm's dividend are compared. 'Known' uses the most recently observed values, 'expected' uses those expected for today given stock and prices, and 'optimal cost' uses expected revenues and profit-maximizing costs C given price beliefs. Even though these rules lead to identical profit-maximizing decisions in equilibrium, their dynamics differ. Three points of interest are $b_R = 0$ (implying $d = \frac{1-\lambda}{\lambda}C$), $b_R = 1 - \lambda = 0.3$ (implying $d = (1-\lambda)R$), and $b_R = 1$ (implying $d = R - C$). This impressively shows how a rule as simple as equation 4 can raise hairy questions when trying to apply it out of equilibrium. The bottommost utility of about 45.5 is achieved by consuming all man-hours as leisure time, with neither production nor any trading taking place. The according source tag is #ExplorationChart. [This figure is enlarged and rotated by 90 degrees in the print version.]

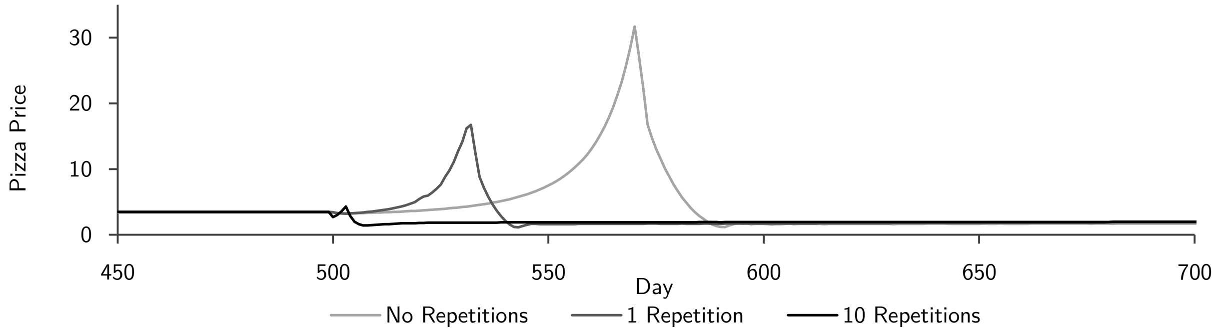


Figure 16: Dry runs of each daily trading session enable faster price discovery. Here, an exogenous preference shock hits on day 500. Source tag: #Tatonnement

4.4. Decentralized Tâtonnement

Léon Walras' well-known tâtonnement process as for example described by Uzawa (1960) can be used to organize dry runs of the daily trading sessions, allowing agents to learn faster and accelerating price discovery. Mandel and Gintis (2014) formally analyze tâtonnement in a decentralized setup with private price beliefs. They find that it covers cases in which the centralized variant does not converge: "it is because our decentralized process has a higher level of granularity that it can converge when the [original] tâtonnement process fails".

Unlike in Mandel and Gintis (2014), where each trading session involves an actual exchange of goods, I stay closer to the original idea and use *decentralized tâtonnement* to organize dry runs of the daily trading sessions. This can be seen as an intraday version of the *reincarnating agents* presented in section 5 as it allows to replay a daily trading session with the agents learning from these repetitions. While one could intuitively expect this to simply accelerate price-finding, these dry runs also allow to tame the medium-term dynamics of the simulation as escalating feedback loops that depend on mispricings cannot fully unfold. Thus, the presented process can also qualitatively stabilize the dynamics with peak mispricings getting less extreme.

I implemented decentralized tâtonnement in a transactional style, i.e. by copying and discarding data instead of trying to explicitely record and undo each trade:

1. All agents and their state including inventories are duplicated - except for price beliefs. The backup copy and the original agent hold same belief object. Thus, adjustments made by the original agent are also visible to the backup agent.
2. Trading takes place as usual, with firms posting offers first and then consumers accepting the ones they like.
3. Firms adjust their price beliefs based on their trading success.
4. If more repetitions are desired, the original agents are discarded and replaced with their backups. The backup agents can take the learned price adaptions into the next iteration, while their inventories are still in the state from before trading.

As a result, prices converge much faster and with less extreme peaks, as shown in figure 16. Generally, having n iterations lets price beliefs converge n times faster, at least when time is measured in simulation days.[27] This is about what can be reasonably expected as the process of repeating the same day resembles a sequence of normal days - except for inventories being reset in the former process. What is

[27] Obviously, the whole process comes at a computational cost, increasing the time it takes to run a single simulation day.

somewhat surprising is that volatility is reduced as well. The decentralized tâtonnement process does not only compress the time axis, but also reduces price peaks and dampens declines in productivity. This makes this method an invaluable tool to exogenously adjust the stability of price-finding to the desired level.

One should note that there are also other limits to how fast prices can converge to an equilibrium after a shock. In particular, using the spending and dividend heuristics discussed in the previous section 4.3, firms will not reach optimal production until their cash holdings have adjusted to the new equilibrium level, a process that naturally takes a few rounds of actual trading as opposed to the fake trading rounds of the tâtonnement process.

Being constructive, Walras' tâtonnement process provides a bridge between classic equilibrium theory and agent-based economics. It also forms the basis of the agent-based simulation done by Gintis (2007). Whether it is a realistic model of price-finding or not is another question that was already discussed early on in economic history. Goodwin (1951), for example, sees Walras' tâtonnement process as a mere mathematical device, whereas Negishi (1962) decidedly disagrees. Being constructive, it certainly is more realistic than non-constructive methods of calculating prices, since reality is constructive as well. Furthermore, even though Walrasian auctioneers are rarely encountered in reality, the Walrasian auction can be interpreted as a negotiation process. In the present decentralized variant in particular, the number of fake trading rounds could thus be interpreted as a way to specify how much effort agents put into negotiating prices with each other.

5. Reincarnating Agents

This section introduces *reincarnating agents*, a method to find self-confirming equilibria in agent-based simulations. Its underlying idea is simple, namely repeatedly running the same simulation and allowing agents to remember observations they have made in previous runs. If lucky, this leads to a stable situation, with agents not altering their behavior any further – at which point a self-confirming equilibrium as defined by Fudenberg and Levine (1993) is found. Under suitable conditions, this self-confirming equilibrium also is a rational expectations equilibrium.

Surprisingly, this idea seems new to agent-based economics, which could be owed to most other authors seeing agent-based simulations as a way to escape rational expectations rather than mimicking them. Furthermore, learning in agent-based economics normally happens endogenously within the model, whereas reincarnating agents learn exogenously, between independent simulation runs. For ease of reference and better differentiation from other learning methods, I propose to name this technique *reincarnating agents*.

5.1. Definition

Reincarnating agents remember local observations from a limited number of previous runs of the same simulation and optimize their behavior accordingly.[28] If an equilibrium is found, subsequent runs of the simulation are identical, and the agents start forgetting about previous non-equilibrium runs. These agents are boundedly rational as they are unaware of what would happen off the equilibrium path. If a sequence of simulation runs with *reincarnating agents* converges, a self-confirming equilibrium is found.

Under suitable conditions, namely "when applied to competitive or infinitesimal agents" (Cho and Sargent, 2008), a self-confirming equilibrium is a rational expectations equilibrium.[29] Also, if a self-confirming equilibrium in an Arrow-Debreu economy is unique, it must be equivalent to the rational expectations equilibrium. This follows from considering that, firstly, every rational expectations equilibrium is a self-confirming equilibrium (Cho and Sargent, 2008), and that, secondly, a rational expectations equilibrium exists in such an economy, as Arrow and Debreu (1954) famously prove.

The question of which observations agents should remember and how they should use the collected information to adapt their behavior is open and left to the architect of the model. However, reincarnating agents should base their decisions on observations they can actually make locally within the model. This stands in contrast to models of bounded rationality such as those presented by Sargent (1993) whose optimizations are based on moments of global variables. For example, basing a decision on one's own average consumption is encouraged, whereas basing the decision on the global average consumption is not. This requirement preserves a clear separation of concerns, reducing the number of dependencies and thereby also the overall fragility of the model.[30] Often, it is possible to reach optimal behavior without remembering every single observation, leading to the interesting question of what constitutes a *sufficient statistic* for optimal behavior.

[28]Note that these repeated simulation runs are *not* a repeated game. Unlike in repeated games, the agent only optimizes for the current run, disregarding that further runs can follow.

[29]In other words: if a single individual cannot change the equilibrium path, none of them will adjust behavior, even if they are rational and could attain a better equilibrium path by collectively changing their behavior at the same time.

[30]Intuitively, one might think that basing all decisions on the same aggregate value is simpler than letting each agent have its own observations. That is true from a global, monolithic viewpoint. However, in an object-oriented model, using already available, local information is always simpler than trying to access information from other agents. A noteworthy approach is also that of the Eurace model by Deissenberg et al. (2008), which contains an explicitely modeled statistical office from which agents can obtain aggregate statistical data.

5.2. Endogenous versus Exogenous Learning

There is a vast number of learning methods discussed in the agent-based modeling literature, with Arifovic (2000), Brenner (2006) and Duffy (2006) providing good examples and overviews. Surprisingly, all the discussed methods are concerned with the agent's endogenous learning *within* a simulation run, as opposed to learning exogenously *across* simulation runs.

Even when other authors choose genetic algorithms, the evolutionary learning is often seen to happen over time within a simulation run, usually with the beliefs being subject to evolution and not the agents themselves. Bullard and Duffy (1999) provide a typical interpretation: "Beliefs are updated via a genetic algorithm learning process which we interpret as representing communication among agents in the economy." Similarily, Arifovic (2000) states: "The population of decision rules is then updated to create a population of rules that will be used at time t + 1." This is not how genetic algorithms are traditionally applied, namely exogenously between independent simulation runs. For example, when training game-playing bots, one would normally let them play full games against each other, with the selection and mutation process happening in between these independent games, as for example Fogel (2001) does with his checkers bot Blondie24. Wright (1995) applies an evolutive algorithm to find an equilibrium in an economic model in that way. However, most existing agent-based economic models let agents evolve within a simulation run, even when it is the agents and not just their beliefs that are subject to evolution. For example, Arifovic (2000) describes an overlapping generations model in which each subsequent generation within a simulation run learns to predict the future a little better.

This design choice by other authors was made with the goal of studying the endogenous dynamics of learning. Lebaron (2002), author of the well-known Santa Fe Artificial Stock Market, writes: "The SFI market could be run as an experiment in the dynamics of a set of rule based trading agents. However, it was always designed to study the emergence of trading patterns as agents learned over time." This stands in contrast to rational expetactions models, that exogenously equip the agents with their theories about the world, regardless of whether the agents actually have enough information to derive said theories. Given the goal of replicating traditional results, my learning method of choice is exogenous, using manually designed adaption heuristics. Also note that in a tournament organized by Rust et al. (1992), the trivial and manually designed heuristic submitted by a student outperformed all the evolutive methods.

5.3. Rational Expectations

> "While rational expectations is often thought of as a school of economic thought, it is better regarded as a ubiquitous modeling technique used widely throughout economics." - Thomas Sargent[31]

Rational expectations provide agents with well-defined beliefs. There are infinitely many arbitrary ways of having bounded rationality, but clear criteria for what constitutes rational expectations. This makes rational expectations an excellent benchmark.

In equation-based models, rational expectations are usually implemented by equating beliefs with the actual outcome of the model, leading to a recursive setup of equations. Thus, Prescott and Mehra (1980) coined the term *recursive competitive equilibrium*. One popular tool to solve equation-based models with rational expectations are Bellman equations, as for example described by Ljungqvist and Sargent (2004). Bellman equations conceptually start with a known future state and then approach the present

[31] Thomas J. Sargent, Library of Economics and Liberty, Rational Expectations, econlib.org/library/Enc/RationalExpectations.html

recursively and backwards in time. Going backwards allows the agents to take their daily decisions under full awareness of what that means for the future. Unfortunately, this approach is fundamentally incompatible with the directional flow of time in agent-based models.

Instead of trying to fix this flaw, many authors of agent-based models distance themselves from rational expectations and emphasize their shortcomings instead. A typical critique can be found on page 7 of Hommes (2013): "Firstly, rational agents are typically assumed to have perfect information about economic fundamentals and perfect knowledge about underlying market equilibrium equations. This assumption seems unrealistically strong, especially since the 'law of motion' of the economy depends on the expectations of *all other* agents. Secondly, even if such information and knowledge were available, typically in a nonlinear market equilibrium model it would be bery hard, or even impossible, to derive the rational expectations forecast analytically, and it would require quite an effort to do it computationally."

Simon (1979) wraps up the trade-off economists face as follows: "decision makers can satisfice either by finding optimum solutions for a simplified world, or by finding satisfactory solutions for a more realistic world." Reincarnating agents help the potentially more realistic agent-based models come closer to the optimum solution. Even if the rational expectations equilibrium is not reached, the goal of reaching a self-confirming equilibrium with agents that each act optimally given their observations is a similarily strong benchmark, providing a clear measure for progress and for evaluating ideas.

5.4. Consumption Smoothing Example

So far, the applied preference shocks have been unanticipated. This allowed to conveniently disregard all intertemporal optimization. Reincarnating agents can adjust their behavior to anticipated shocks, which would not be possible in agent-based models with endogenous learning. By definition, endogenous learning within a simulation run cannot prepare agents for exogenously applied shocks. The presented example of consumption smoothing agents seems to be the first agent-based model with anticipated shocks.[32]

Starting with the production economy from section 3, the model is simplified to only one firm type 'pizzeria' and one consumer type 'Italian'. The recommended dividend heuristic from section 4.3 is applied. The tâtonnement process from section 4.4 is enabled with up to 10 iterations, accelerating price-adjustment as that is not the dynamic of interest. The simulation runs for 1000 days, with an exogenous shock doubling the preference for pizza from $\alpha = 6$ to $\alpha = 12$ on day 500. It starts at day -100, allowing the economy to warm up before measurements start. Consumers observe their consumption over time, with each incarnation of consumers trying to adapt daily savings before the shock such that they have enough reserves to double their consumption after the shock. They do not make any second-order considerations. The source tag of this configuration is #Smoothing.

During the first run, the shock comes at a complete surprise, making the consumers work inefficiently hard afterwards in order to satisfy their sudden craving for more pizza. The following reincarnation uses the observed average consumption before and after the shock to calculate the daily savings necessary to sustain a doubling of consumption after the shock. However, it neglects second-order effect, namely that saving some of the acquired pizza leaves the consumer a little hungrier, making the consumer work a little longer to buy a little more pizza. Analogously, the consumer is a little less desparate and works a little less hard after the shock. With each reincarnation, savings are slightly readjusted

[32]While there are examples of basic agent-based models with exogenous learning (e.g. Wright (1995)), I did not find any such model that includes anticipated shocks.

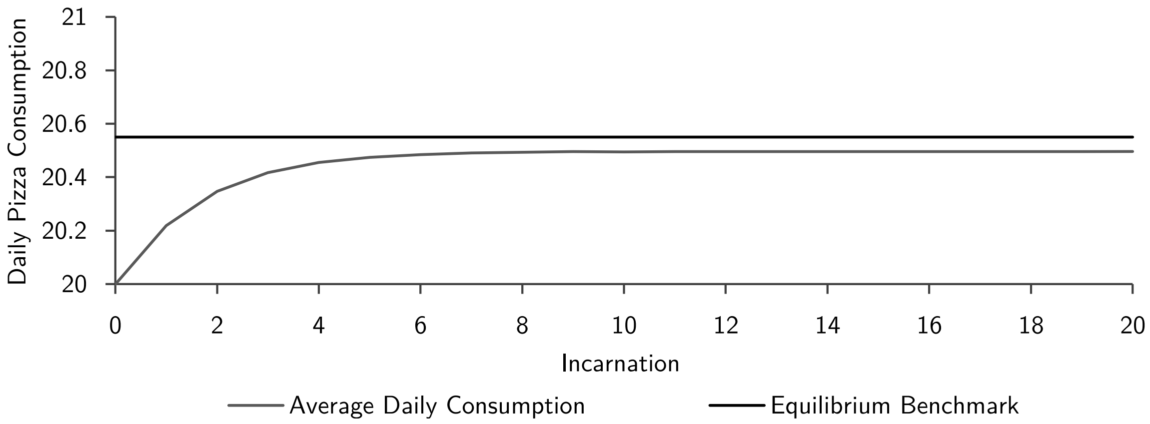


Figure 17: Daily production quickly converges after a few simulation runs to about 0.26% below the efficient benchmark.

upwards, converging towards the rational expectations equilibrium.[33] In equilibrium, the consumer works equally hard before and after the shock, with production and prices consequently also flattening. The resulting price chart for pizza is shown in figure 18.

Figure 17 takes a closer look at how average daily production evolves over time and how close it comes to the equilibrium benchmark. Increasing the number of simulation days as well as increasing the number of tâtonnement iterations improves accuracy further to 0.1% below the benchmark. Both is good enough to show that the concept of reincarnating agents can work.

5.5. Mixing Anticipated and Unanticipated Shocks

The given example proves that reincarnating agents can work as a method to approach a self-confirming equilibrium. This method works under heterogeneity and can deal with any number of local and global shocks, given that the architect of the model equips the individual agents with suitable decision heuristics. In the case of multiple anticipated shocks, all can be applied from the beginning. However, when mixing them with unanticipated shocks, a more elaborate procedure is necessary:

1. All anticipated shocks are included and the simulation repeated until a self-confirming equilibrium is found.[34]
2. The first unanticipated shock happening on day t is included in the simulation's configuration.
3. The simulation is repeated partially, starting on day t, until a new equilibrium is found.
4. If there are more unanticipated shocks, the next one is included, t adjusted forwards, and step 3 repeated.

[33] To calculate the rational expectations equilibrium, consider that given log-utility, the optimal choice is to work equally hard all the time and to consume twice as much after as before the shock, letting the two-period utility function collapse to simple $U(x,h) = (6+12)log(x+1) + 12log(2) + (14+14)log(25-h)$ with production function $x = (10h)^{0.7}$, resulting in an optimal average consumption of $x \approx 20.55$.

[34] Note that there is no guarantee that this will ever happen.

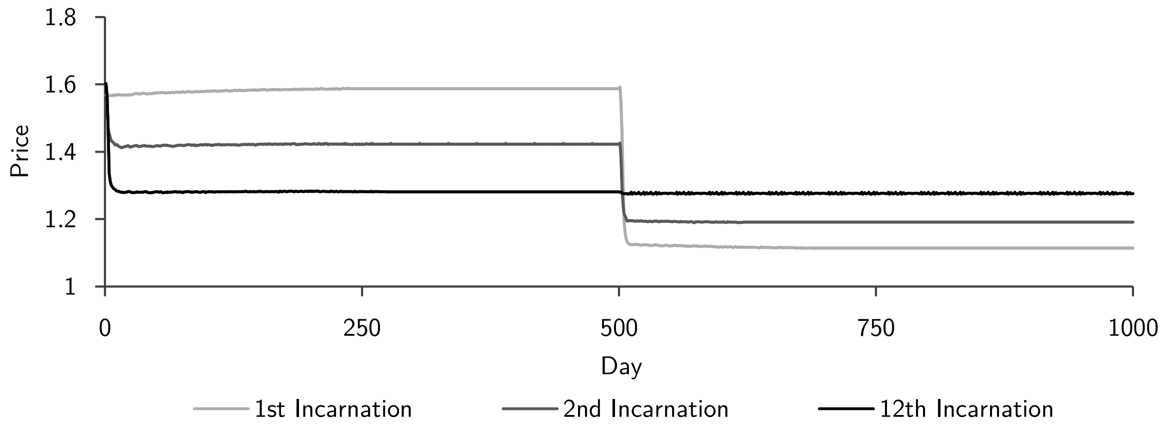


Figure 18: With each reincarnation, consumers adapt better to the preference shock on day 500, finally settling on a self-confirming equilibrium with smooth production and prices. This equilibrium coincides with the rational expectations equilibrium. Somewhat counter-intuitive, prices in the first run are lower after the shock, when more pizza is demanded. This is owed to the higher production. With money supply staying constant and trade volumes increasing, prices must fall according to the Fisher (1922) equation $MV = PQ$.

5.6. Limitations

5.6.1. Computational Costs

The inclusion of additional *anticipated* shocks increases the computational costs of applying the consumers' decision heuristics, whereas the inclusion of additional *unanticipated* shocks increases the number of necessary simulation runs. Assuming that both these dependencies are linear, and given that the presented example takes about one second per run, or ten seconds for ten reincarnations, a quick back-of-the envelope calculation indicates that this would still allow to conveniently include hundreds of shocks, but not thousands if one wants to be able to find the self-confirming equilibrium within a few hours on an average computer.

This compares favorably to general numerical solution methods of equivalent equation-based models, whose computational costs usually grow non-linearly as the number of variables and equations increases, unless model-specific optimizations are identified and applied. As noted in section 3.5.3, both approaches are eventually bound by the same theoretical limits. Having an agent-based model that finds the solution quickly indicates that there must also be an equally fast method for solving the equivalent equation-based model. However, in certain cases, running the agent-based simulation might be that method.

5.6.2. Intertemporally Consistent Prices

A limitation of the presented model is that relative prices before and after shocks might leave potential for intertemporal arbitrage. In the given example, this happens not to be the case as prices are flat in equilibrium. It would also work out correctly when flipping preferences to $\alpha = 12$ first and $\alpha = 6$ after the shock - in which case no consumption smoothing would take place as pizzas cannot be borrowed from the future. Having low prices first and high prices later would seem like a good arbitrage opportunity. However, the only good an agent could buy with the profits is pizza again, neutralizing the trade in real terms. In more complex configurations, these considerations might not hold any more.

To address this, a mechanism for the intertemporal adjustment of prices would need to be introduced. Ideally, this would ensure that profit-maximizing in real terms is equivalent to profit-maximizing in nominal terms.

One potential way of making prices intertemporally consistent could be the introduction of a reincarnating arbitrage trader who buys in cheap periods and sells in expensive periods. Maybe, such a speculator could even take over the role of storing the ideal amount of goods for consumption smoothing - with all coordination happening through the price system and thereby encapsulating this concern in a specifically designed agent specialized on the storage of goods. A few preliminary attempts of doing so unfortunately failed due to price volatility in connection with that arbitrageur first pushing its money into the economy and later pulling it out again.[35]

A further complication is posed by the fact that equilibrium models of markets usually do not contain money. Thus, they cannot serve as a benchmark for nominal prices. The classic Arrow-Debreu opens only once to fix all present and future trading. Money is neither required as a means of exchange nor as a store of value. Overlapping generations models that include money as a store of wealth only work by imposing that the young must accept the money of the old, when in fact they would be better off using their own money and letting the old starve. (Samuelson, 1958) Relative nominal price levels in such models are exogenously imposed and not useful as a benchmark either.

One potential way of defining intertemporal price consistency could be to require the marginal utility of what money can buy to stay constant. However, this approach suffers from the question of which consumer or group of consumers should serve as a reference in a world with heterogenous preferences. Addressing intertemporal nominal price stability in agent-based simulations could be a challenging goal for future research.

5.6.3. Endogenous Life-Cycles

The method of reincarnating agents in its presented version requires agents to appear exogenously. In a model with endogenous births and deaths, it would be hard to define which agent's observations should be passed on to which agent in the next simulation run. Maybe the same agent is not born any more in the next run, and other new agents from different parents appear instead. Whether the method of reincarnating agents could somehow be adapted to simulations whose agents have endogenous life-cycles remains an open question subject to future research.

[35] In an other failed attempt of indirect consumption smoothing, the firms were doing the saving instead of the consumers themselves, an idea which cannot work as the consumers have no way of distinguishing between the firms putting goods aside for later and the firms simply having a reduced productivity - two scenarios that have a different optimal consumer behavior.

6. Stock Market

This section adds a stock market on which all firms of the production economy are publicly listed. Instead of distributing dividends equally among all consumers as before, dividends are now distributed to the shareholders in accordance to the number of shares they hold. Since the static case with immortal consumers does not require any trading, consumers are made mortal. They are continuously added into the model as time passes by, working during the first part of their life and spending the second part in an exogenously imposed retirement. Thus, they must save and invest in the stock market when young in order to smooth consumption over time.

This part of the model is much more explorative, somewhat detached from existing theory, and lacking strict quantitative benchmarks. Instead, the impact of individual changes in micro-behavior on the macro-result is tested, comparing the outcomes with earlier outcomes of the same model. This differnatial testing allows to playfully attain a sense for what is going on. Informally, this is done by looking at charts. More formally, correlations between relevant metrics are ranked and commented, and in one case statistical properties compared to the real world.

The first three sections specifiy the behavior of the three agent types that are active on the stock market, namely consumers, market makers and investment funds. In section 6.4, market makers are added and shown to qualitatively change price formation. A periodically fluctuating birth rate is used to make its influence on other variables better visible. In section 6.5, the savings rate is varied. From Section 6.6 on, the birth rate is flattened again, allowing to focus on the chaos caused by emerging circular ownership structures between funds. Finally, section 6.9 reflects on the gained insights.

6.1. Saving Consumers

The mortal consumers all live for exactly 1000 days, the last 400 days of which are spent in retirement. In retirement, they consume their whole daily man-hour endowments as leisure time. At birth, they do not own any shares yet. They work from day one and follow a simple strategy of observing their daily nominal spendings and investing 40% of that into the stock market. Similar savings plans are also often recommended and applied in the real-world, an example being the Swiss 'three pillar' retirement system that allocates between 17% and 50% of income.[36] Furthermore, it is dynamically robust as the absolute amount of savings does not fluctuate more than spendings. The dynamics could pose a problem in systems where consumers try to reach a savings target of for example 400 times the daily spendings. There, small fluctuations in daily spendings lead to large changes in the target savings amount and are therefore more likely to destabilize the system.

Retired consumers stop working and try to sell a fraction $1/d$ of their total savings for consumption on each day, with d being the number of days left to live including the current one. On their last day, they sell everything they have left to buy and consume consumption goods.[37]

Consumers are equiped with two different heuristics for executing their daily stock purchases. In *equal weight* mode, consumers randomly select a company with equal probability. In *index* mode, the

[36] The amount depends on age, the employer, and the employee's choice. Some flows into an unfunded 'pay-as-you-go' system, and some into specifically regulated retirement funds. All in all, the 40% are a reasonable first approximation. Source: www.bsv.admin.ch/kmu/ratgeber/00848/00859 and related sites.

[37] Note that this simplified behavior is not perfectly optimal as it does not take dividend income into account. For example, when consuming shares worth 10$ over the course of two days, this heuristic would let the consumer sell 5$ worth of shares each day, but leading to consumption worth 6.0$ and 5.5$ when the dividend yield is 10%. A strictly better result would be achieved by selling only stocks worth 4.5$ on the first day, leading to consumption worth 5.5$ and 6.05$ instead.

random selection is weighted by market capitalization. The *index* strategy resembles the often recommended strategy of investing in index funds, which are usually weighted by market capitalization. As shown later, this can make a difference for the market dynamics. When selling, consumers make no selection and sell a fraction $1/d$ of each position in their portfolio.

6.2. Market Makers

Many models of financial markets implicitely assume infinite liquidity, even though they acknowledge that this is not entirely correct. For example, Samanidou et al. (2007) writes: "Given a more or less continuous revaluation of the portfolio structure, the floor is therefore (quite) safe," implicitely assuming that prices are not affected if positions are liquidated fast enough. without affecting the price. Explicitely modeling market making forces one to drop such assumption and to adopt an approach of limited liquidity.

The role of the market maker agent is to provide that liquidity and to find market prices in a process similar to how the firms find out about the market prices of goods. In particular, each market maker maintains two price beliefs for each stock with a minimum spread of 1% in between. Furthermore, they adjust price beliefs by at most 10% per day, capping volatility. The market maker agent is inspired by real-world market makers, whose job is to permanently post bids and asks with a contractually defined spread.[38] In return, they get privileged access to the markets and are sometimes even paid by the listed company for doing the market making. They usually do not do any fundamental analysis, but can earn money proportional to the spread as investors buy and sell from them.

In the model, the market maker agents are the only ones that post limit orders to the market, ensuring that all trading goes through them. There are multiple market makers in competition with each other, making sure that there is competitive pressure to keep spreads tight. Their behavior is designed such that they can never run out of a stock once they own it. This is achieved by always offering a fraction of 5% of their holdings on the ask side of each stock. Whenever such an ask is filled, the according price belief is adjusted upwards exponentially as described in section 3.3.[39] In the case of sustained demand, price beliefs can rise faster than 5%, ensuring that the nominal value of the posted ask stays significant despite the declining number of offered shares.

On the bid side, the market maker maintains a second price belief for each stock. The bid price belief can move independently of the ask price belief, allowing spreads to increase in times of strong concurrent demand for both, buying and selling. In times of no or minimal trading activity, the two beliefs move towards each other until they reach the minimum allowed spread of 1%, which also ensures that the two beliefs never cross. Each day, the market maker proportionally allocates the available cash (but at most 1000$) to bid for all available stocks at the current bid price belief. For example, if it has 1000$ cash and there are 20 companies, it posts 20 bids of size 50$.

Market makers are listed companies themselves and are allowed to buy each others shares, except their own. They start with an endowment of $e = 1000\$$, and distribute $d = max(0, (c - e)/3)$ as dividend every morning, with c being their current cash level. In the long run, this is equivalent to distributing daily profits.[40] The income of market maker stems from two sources, trading as well as dividends from the companies in their portfolio. Market makers sometimes trade among themselves

[38] For example, on the Swiss SIX exchange, the maximum allowed spread is 2% for exchange traded products. See six-swiss-exchange.com/participants/trading/etps/market_making_en.html.

[39] While exponential search is helpful here, sensor prices are not necessary.

[40] For constant profits $\pi \geq 0$, cash follows $c_{t+1} = c_t + \pi - \frac{c_t - e}{3}$. Setting $c_{t+1} = c_t = c^*$ results in $c^* = 3\pi + e$ and thus $d = \pi$.

in case their beliefs do not match, which makes their beliefs gravitate towards each other via the price adaption mechanism.

The behavior of the market maker has been found by trial and error with the goal of finding the simplest competitive heuristic that works well in the simulation. This behavior does not maximize profits. Regardless of whether nominal or real prices are used, maximum profits would be attained by having two market makers owning each other and moving as much money as possible back and forth between them as dividends. In such a scenario, the profits would only be limited by how fast money can flow, which is obviously not something that brings real value to the economy. Yet, it is not obvious to me how to elegantly resolve this problem.

6.3. Investment Funds

So far, the model contains no forces that pushes stock prices towards their fundamental value. This is the role of the investment fund agents. Every day, these agents try to sell shares with a low dividend yield and buy shares with a high dividend yield, thereby optimizing there income. Excess money is returned to the shareholders as dividends. Being listed companies themselves, investment funds are allowed to buy each others' shares, but not their own.

After the market makers have posted their limit orders, all other market participants – namely consumers and funds – manage their portfolio in random order. The investment funds do so by ranking all companies by dividend yield, calculated by dividing a moving average of past dividend payments by the current ask price. If there is no ask left in the market for a company, that company is skipped. The worst fifth of the companies found in that ranking are sold if present in the portfolio. The best fifth of these companies are bought, investing at most 1000$.

Furthermore, the investment fund keeps track of its inner and outer value. The outer value is simply its market capitalization. The inner value is the market value of the portfolio including cash. In case the inner value is more than 50% above the outer value, the shareholders can obviously be made better off by returning capital. Thus, purchases of further shares are not allowed under this condition, freeing up cash for dividend payments. Conversely, as long as the outer value is more than 50% above the inner value, all selling of shares is stopped. Dividend payments are calculated the same way as for the market makers.

6.4. Sine Births

This configuration shows how demographics affect all other economic metrics. There are no major surprises, making it a good base case for further exploration. The birth rate on day d is given by the sinusoid

$$n\frac{sin(2\pi d/p) + f}{pf}$$

with a period of $p = 1500$, $n = 100$ births per period, and flatness $f = 2.0$, leading to the population dynamics depicted in figure 19. As a result, the dependency ratio $r_{dep} = \frac{n_{retirees}}{n_{workers}}$ fluctuates between $0.3 < r_{dep} < 1.4$, with the overall average being $\bar{r}_{dep} = 2/3 = 400/600$ as consumers spend 400 of their 1000 days in retirement. A stable real-world population with life-expectancy 82 also spends 60% of its time working, namely from 15 to 65 according to the standard definition of the dependency

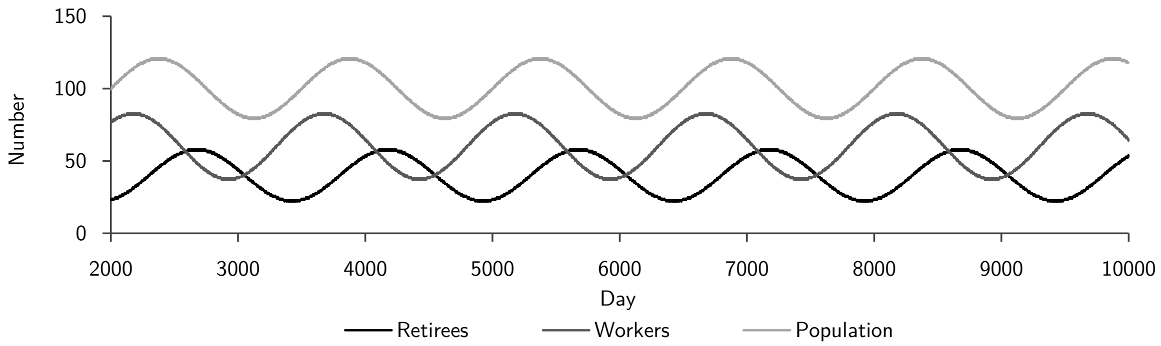


Figure 19: Population dynamics with a sinoid birth rate, a life-expectancy of 1000 days, and retirement from age 600.

Variable 1	Variable 2	Correlation	Comment
p_{food}	p_{med}	0.92	All nominal prices are strongly correlated.
p_{food}	p_{hours}	0.95	
p_{hours}	p_{med}	0.84	Medicine prices correlate to a smaller extent.
p_{food}	v_{food}	-0.88	Prices are low when volumes are high.
v_{food}	$n_{workers}$	0.98	More workers produce and eat more.
v_{med}	$n_{workers}$	0.54	More workers also produce somewhat more medicine.
p_{food}/p_{index}	p_{hours}/p_{index}	-0.95	Strong, but artifact of index construcion.

Table 4: Correlations

ratio.[41] Having a cyclical scenario without long-term growth is well-suited for visual impact-analyses, as the same cyclicality can be spotted in dependent variables. The source tag of this configuration is #Savings10.

These workers and retirees live in a world with only three goods, the first of which is man-hours. The other two goods are food and medicine, this time fully perishable so there is no point in storing them, allowing to conveniently disregard any savings decisions in terms of real goods. While young and working, consumers have preferences $\alpha_{young} = 9$ for food and $\beta_{young} = 1$ for medicine. The older retirees have preferences $\alpha_{old} = 5$ for food and $\beta_{old} = 5$ for medicine, assuming a decreased appetite and reduced health in a gross simplification. There are 10 firms for each product with Cobb-Douglas production as in section 3.1, except that there is only one type of labor with intensity $\lambda = 0.7$, which also reprents the returns to scale.

This all results in the prices shown in figure 20. While one might initially expect food and medicine prices to follow different patterns due to changes in demand, nominal prices are strongly correlated, as shown in correlation table 4. These synchronous price movements are primarily driven by trade volume v shown in chart 21, which is high when prices are low and vice versa. This is a consequence of the constant money supply and in accordance with the Fisher (1922) equation $MV = PQ$, saying that – ceteribus paribus – prices P and trade volumes Q are indirectly proportional.

[41]The standard definition is to count those aged 15 to 65 as working, and everyone else as dependent. Even though the ratios have similar values, the dependency in the model is qualitatively different as consumers are born as adults. Real-world dependency ratios in 2014 were between 0.17 in the United Arab Emirates and 1.13 in Niger according to the world bank (data.worldbank.org/indicator/SP.POP.DPND).

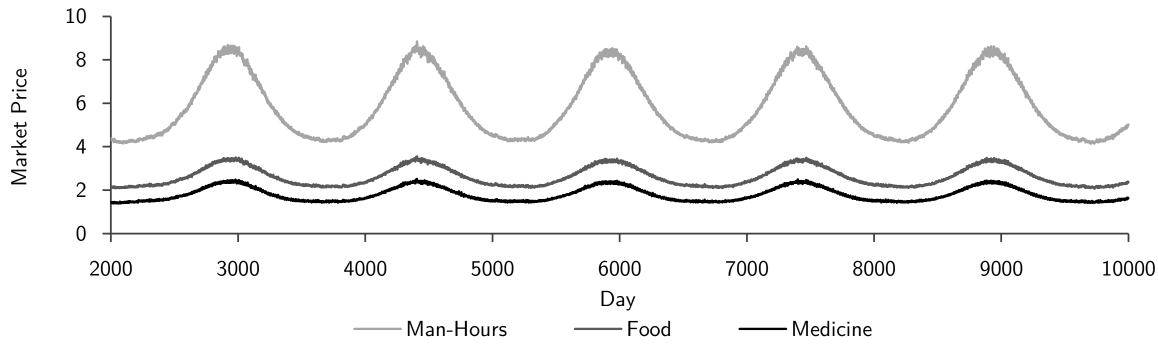


Figure 20: Nominal prices correlate strongly with each other.

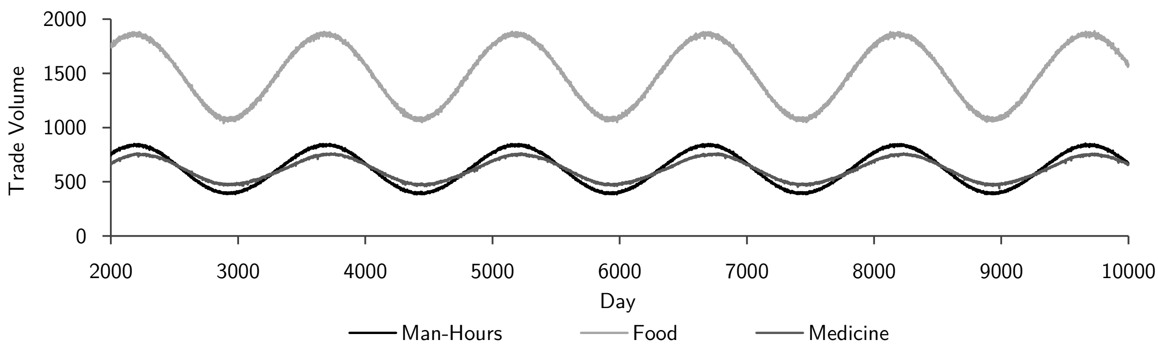


Figure 21: Trade volumes move in opposite direction to prices.

$$p_{index} = \frac{p_{hours}v_{hours} + p_{food}v_{food} + p_{med}v_{med}}{v_{hours} + v_{food} + v_{med}}$$

In order to disentangle price movements from each other, a volume-weighted price index is constructed and real prices calculated, dividing nominal prices by the index. In real terms, prices of man-hours and food mirror each other in a strong negative correlation (-0.95). This is an artifact of these two goods dominating the price index. It basically says that when food is expensive relative to man-hours, man-hours must be cheap relative to food – an unhelpful tautology.[42] However, the price index still helps revealing that the medicine price and the food price do not peak at the same time in real terms as depicted in figure 22.

In order to save for retirement, consumers buy stocks on the stock market. For now, there are market makers, but no investment funds. Thus, there is no market actor to drive stock prices towards their fundamental value. Nonetheless, all stock prices tend to stay close. They move with a correlation of 0.99, as shown in table 5. This is owed to the equal-weight buying strategy of randomly selecting a listed company with equal probability. Since the amount spent on each stock does not depend on that selection, it is more likely that a given purchase is big enough to push a low price upwards than that it is big enough to push a high price upwards. On the selling side, the analogous effect pushes relatively high prices downwards faster as retirees always choose to sell a given percentage of each portfolio position, regardless of price.

[42]Note that $corr(\frac{A}{A+cB}, \frac{B}{A+cB}) = -1$ for random variables A and B, and constant $c > 0$. When disregarding medicine, these A and B resemble p_{food} and p_{hours} divided by the weighted price index $(p_{food} + vp_{hours})/(1+v)$ with $v = \frac{v_{hours}}{v_{food}}$.

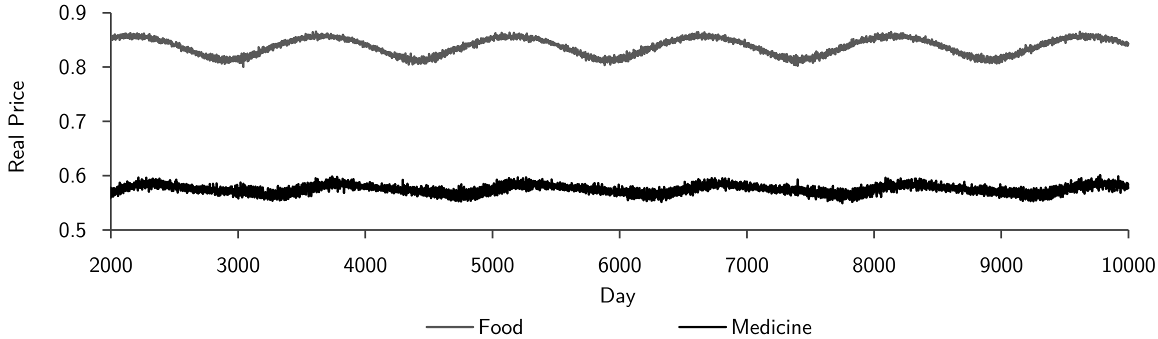


Figure 22: Charting real prices reveals that lows and highs of food and medicine prices are not perfectly synchronized.

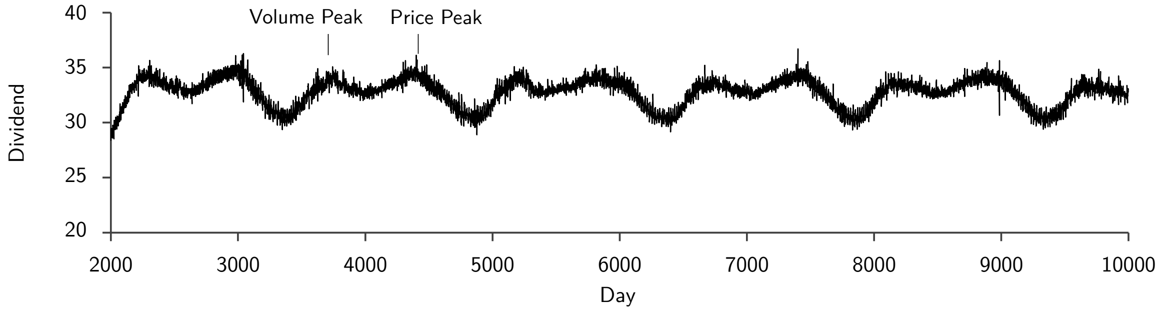


Figure 23: Dividends of medicine firms show a repeating camel pattern with a first peak driven by high sales volume, and a second peak driven by high prices.

Variable 1	Variable 2	Correlation	Comment
p_{FOOD}	p_{MED}	0.99	All stocks are strongly correlated.
p_{FOOD}	p_{MARK}	0.99	Also the shares of food firms and market makers.
p_{MARK}	p_{MED}	0.99	And the shares of market makers and medicine firms.
$ln(p_t/p_{t-1})$	u_t	0.30	Daily log-returns correlate with daily experienced utility.
p_{ALL}	v_{ALL}	-0.64	Fewer shares are traded when they are expensive.
p_{ALL}	$n_{workers}$	0.47	Stock market is high when there are many buyers.
p_{ALL}	$n_{retirees}$	-0.42	Stock market is low when there are many sellers.
p_{ALL}/d_{ALL}	n_{work}/n_{ret}	0.89	Price-earnings and inverse dependency ratio.
log returns	net inflows	0.86	Net inflows determine returns (here over 31 days).

Table 5: Stock market correlations, with ALL denoting the index and v volume.

Chart 24 shows how the stock market develops over time. It exhibit the same 1500-day cycle as the other metrics, with a relatively low advance and steep declines. In the absence of speculators, the stock market is driven by the difference between inflows and outflows, which are shown in figure 25. Whenever the workers invest more than the retirees divest, the stock market goes up and vice versa. At first sight, this sounds like classic price clearing. However, here it is the change of the prices that are being driven by the investment amount, not the prices themselves!

When a buyer willing to invest $i = 100\$$ encounters a seller willing to sell $n = 10$ shares, they are usually expected to settle at a price $p = i/n = 10\$$. This is not what happens on the simulated stock market. Here, the buyer acquires n_{buy} shares from the market maker at price p, and the seller sells $n_{sell} \neq n_{buy}$ shares to the market maker at price $p(1-s)$ with s being the spread. Afterwards, the market maker adjusts the price p upwards or downwards a little, depending on whether p was too low or too high.[43] The market maker's price belief is too low and too high equally often, neutralizing capital gains and losses over time.

These market maker agents turn out to be a device to connect returns instead of prices to money flows. In mathematical terms, the observed law of motion for the average price p with stock market inflows i and outflows o resembles

$$\frac{dp}{dt} = c(i-o) = c(i-np) \tag{9}$$

with c being a configuration-specific constant. The special case $dp/dt = 0$ implies the traditional market clearing rule $p = i/n$. Including a market maker allows inflows and outflows to differ, with the market maker absorbing the difference and thereby qualitatively changing price formation.[44] Averaging the data over $s = 30$ days and testing the postulated law of motion in regression

$$p_{t+s+1} - p_t = c\sum_{d=0}^{s}(i_{t+d} - o_{t+d})$$

results in $c = 0.001$ with a t-statistic of 138 and an excellent $R^2 = 0.74$.[45] Apparently, investing 1000$ lifts the average stock price by one dollar, which is a 3500$ increase in total market capitalization (there are 35 companies with 100 shares each). The reported R^2 corresponds to a correlation coefficient of 0.86, which is the metric used in the following comparisons and calculated on the basis of log return r as shown in equation 10 for simpler handling.

$$corr(\sum_{d=0}^{s} r_{t+d}, \sum_{d=0}^{s}(i_{t+d} - o_{t+d})) \tag{10}$$

For $s = 0$, the measured correlation is 0.11, for $s = 3$ it is already 0.44, and for $s = 30$ it is 0.86. This confirms that the ups and downs of the stock market in this scenario are driven by the investments and savings of the consumers.

Geanakoplos et al. (2004) predict price-earnings ratios to be proportional to the ratio of middle-aged to young, whereas in their model it is the middle-aged who do the saving. The simulation is in line with this finding, exhibiting a correlation of 0.89 between the price-earnings ratio of the index and the worker-retiree ratio. In contrast, Poterba (2004) concludes that "these empirical findings provide modest support, at best, for the view that asset prices could decline as the share of households over the age of 65 increases." It might be worthwhile for future research to investigate this in a simulation with more realistic parameters such as real-world population structures.

[43] This is a simplification. For the actual adaption algorithm with separate price beliefs above and below the spread, see section 6.2.

[44] In the simulation, the market maker's holdings vary between 18% and 27% of all shares as the stock market goes up and down.

[45] Like in the charts, the first 2000 days of the simulation are skipped.

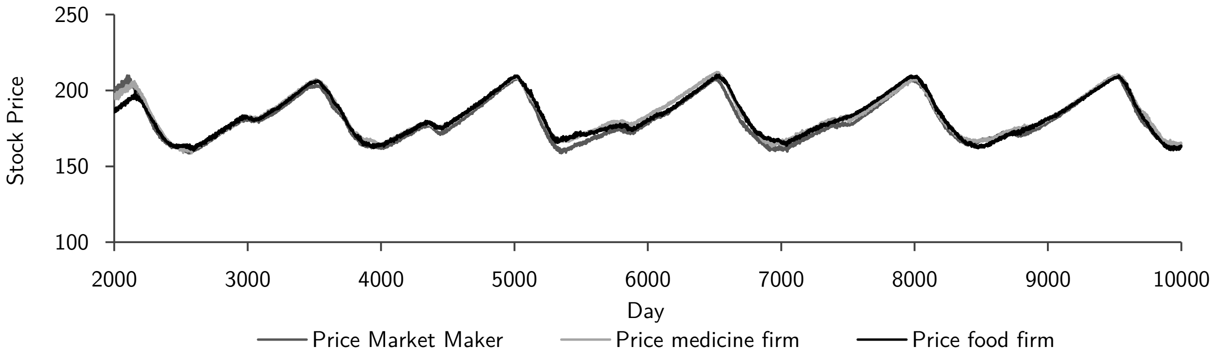


Figure 24: Stock prices climb slowly and fall quickly. Volatility is low compared to the prices of goods.

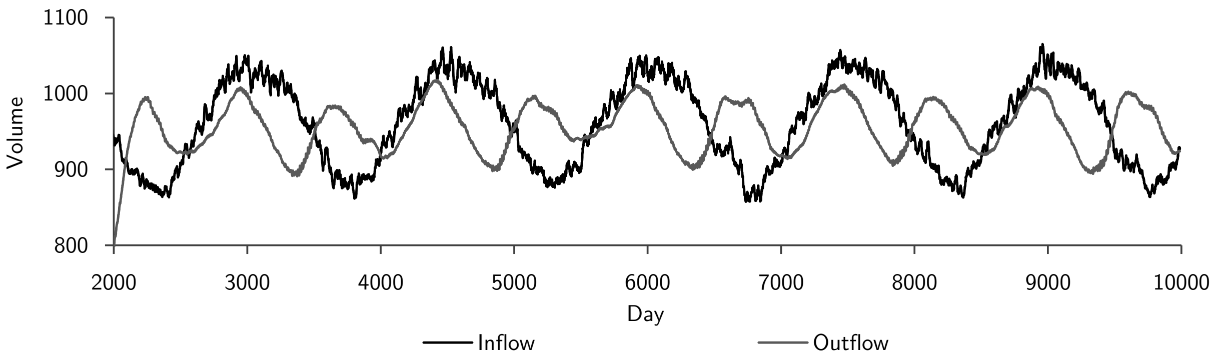


Figure 25: As long as workers invest more money into stocks than retirees are divesting, the stock market rises. The difference between inflows and outflows has high explanatory power for log returns. Under classic Walrasian market clearing without market makers, inflows and outflows would always be identical. Here, this only holds when summing over a full cycle.

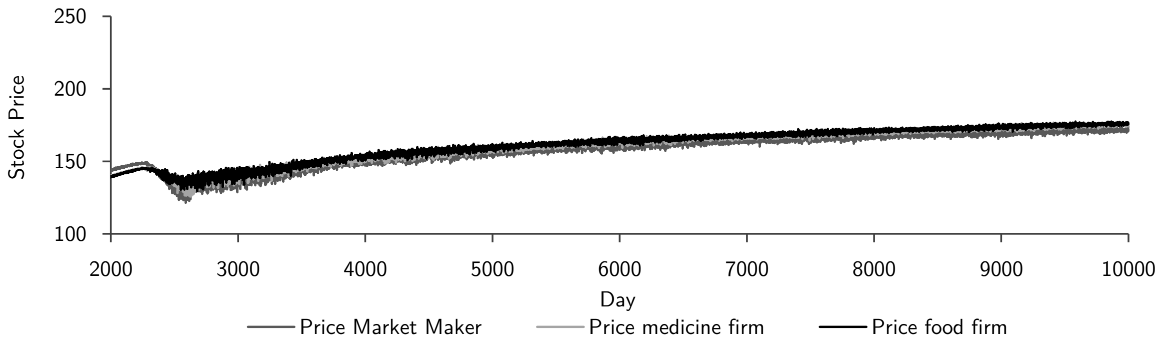


Figure 26: When consumers save only half as much, market makers can dominate the market and absorb all changes in in- and outflows. The price curve smoothly moves towards what looks like an asymptotic equilibrium. The volatility around day 2500 is an artifact of the initial conditions. Source tag: #Savings05

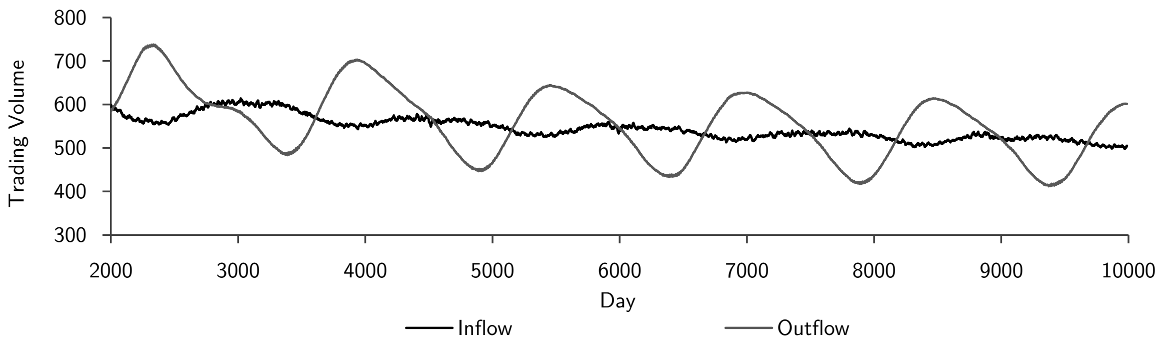


Figure 27: The fund flows are about half as high when consumers save half as much, but also differ qualitatively in the absence of stock market volatility.

6.5. Varying the Savings Rate

Given the base case specified in the previous section, the savings rate is varied to see how it impacts the stock market and how robust the aforementioned results are.

Reducing the savings rate leads to a decline in the correlation between net inflows and log returns. A reduction to 75% of the optimal level, for example, lowers the 31-day correlation from 0.86 to 0.64. At some point, a qualitative change happens and the stock market behaves fundamentally different, as shown in figure 26. At this point, the relation between inflows and log returns does barely hold any more (correlation 0.10 over 31 days) and the fund flows shown in chart 27 fail to shape stock market prices. In this configuration, the market maker dominates the market by holding about 50% of all shares. With such power, the market maker can absorb all fluctuations in fund flows and the price curve apparently follows a smooth path towards an asymptotic equilibrium.

Increasing the optimal savings rate from the base case by 50% leads to an increase of the correlation between net inflows and log returns, but only in the short run. Whereas the 31-day correlation declines from 0.86 to 0.62, the 4-day correlation grows from 0.44 to 0.47 and the daily correlation from 0.11 to 0.16. The correlation of price-earning ratio to worker-retiree ratio declines somewhat to 0.80.

With the market makers having relatively less weight, the savings flows can shape price movements more strongly, making both more volatile as shown in figures 28 and 29. These charts also reveal the

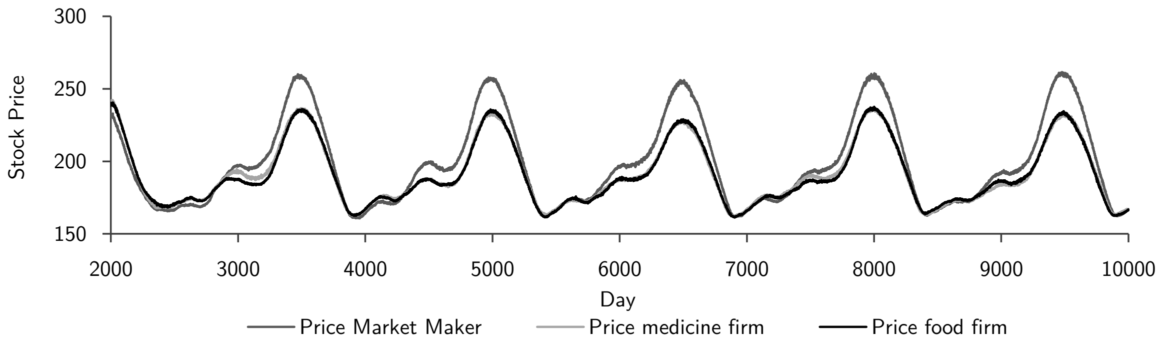


Figure 28: When increasing the savings rate by 50%, market makers cannot smoothen prices as much any more and volatility increases. #Savings15

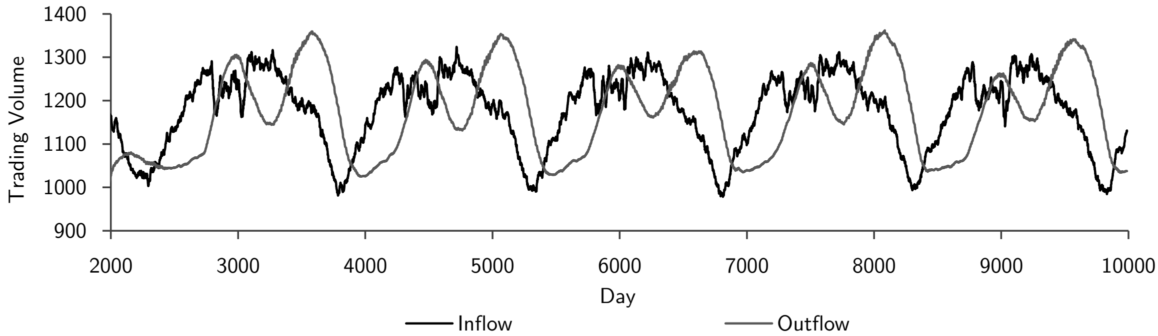


Figure 29: With increased savings, the difference between inflows and outflows still impacts returns, but not as consistently any more as in the base case.

limits of the previously postulated law of motion for prices, with the stock market slightly dipping at a point in time where flows still would suggest positive returns.

6.6. Buyback Escalation

In an attempt to bring the stocks in line with their fundamental value, investment funds with a value strategy are added to the model. However, it turns out that these funds can destabilize not only the stock market, but also the real economy. After showing what happens to the economy from the previous section when adding value investors, a configuration with a simplified real economy is introduced in order to better isolate financial market dynamics.

Putting consumers into *index* mode as defined in section 6.1 and adding five fundamentalists as defined in section 6.3 successfully moves the prices of the food firms and the medicine firms towards their relative fundamental value, namely food firms being priced three times as high as medicine firms as they also emit three times as much dividends.[46] However, the market as a whole is destabilized, with the share price of the investment funds escalating to infinity (chart 30) as the circular ownership structure from graph 32 emerges.

[46]With consumers in *equal weight* mode, food firms shares only reach twice the price of medicine firms, but the stock price of the funds still escalates.

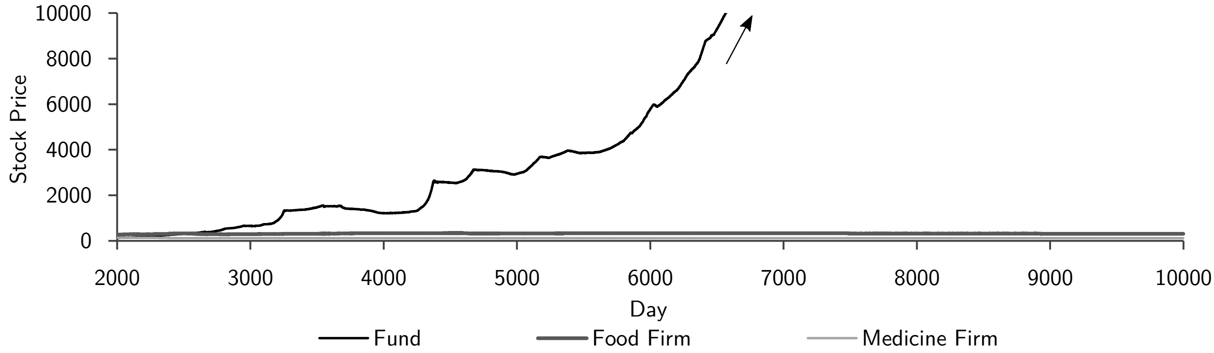


Figure 30: With fundamentalist funds, the relative prices of food firms and medicine firms is in line with their fundamental value. The stocks of the funds, however, go through the roof. The somewhat erratic price of the market makers is not shown.

One should note though that an infinite price per share can be justified when the free float goes towards zero and all the other shares are owned by the company itself. For example, a company worth 100$ with ten outstanding shares should have a share price of 10$. However, if eight of those ten shares are owned by the company itself, the price of a share jumps to 50$. With continuously divisible shares like in the simulation, the marginal price of a single share can thus go towards infinity as the company owns more and more of its own shares (directly or indirectly).

In an efficient market, share buybacks should have no effect at all on price. If a company spends 10$ to repurchase and destroy[47] one outstanding share worth 10$, the value of the company and the market capitalization both decline by 10$, with the remaining shares having the same price as before. In mathematical terms, a company worth v with correct market capitalization $m = v$ remains correctly priced after buying or selling one share at market price p:

$$\frac{m}{v} = \frac{m+p}{v+p} = \frac{m-p}{v-p} = 1$$

However, if the company has an overvalued market capitalization $m_o > v$, a buyback will amplify that overvaluation as $\frac{m_o-p}{v-p} > \frac{m_o}{v}$. Analogously, an undervaluation is also amplified by a buyback as

$$\frac{m_u - p}{v - p} < \frac{m_u}{v} \tag{11}$$

for $m_u < v$. Share buybacks amplify mispricings, and if an ongoing buyback amplifies an undervaluation faster than the market adjusts prices upwards, the price per share can be pushed to arbitrary heights. Figure 31 illustrates this effect in a causal loop diagram. This is what drives the price escalation in the simulation. Once a significant level of self-ownership is reached, circular dividend flows additionally incite the escalation.

Amplifying mispricings is a recurring theme in agent-based simulations, be it through chartists (Frankel and Froot, 1990), market sentiment (Lux and Marchesi, 1999), endogenous learning (Lebaron, 2002), leverage (Thurner et al., 2012), or a mixture of aforementioned factors (Hommes and in 't Veld, 2014). Amplifying mispricings through direct or indirect buybacks seems a novelty. This is partially owed to other models rarely ever having explicit ownership structures. And even if they have explicitely modeled stock ownership, investors are usually not listed companies themselves, ruling out circular ownership structures by design.

[47] In practice, it is irrelevant whether the purchased share is destroyed or not. From the outside, repurchasing and selling is indistinguishable from repurchasing, destroying and reissuing.

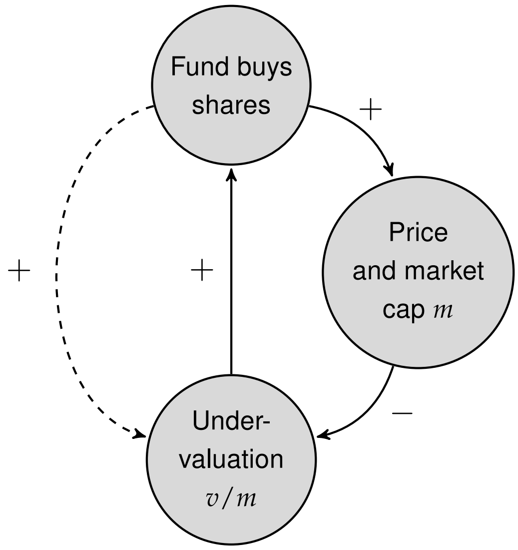


Figure 31: The causal loop diagram of the buyback escalation. The dashed dependency appears as soon as a fund buys its own shares, directly or indirectly. It creates a reinforcing feedback loop that increases the undervaluation $v/m > 1$ through the mechanism from 11. If the reinforcing loop is stronger than the balancing one, the price escalates.

Looking at the impact on the real economy, nominal prices fall continuously as the funds hoard more and more money, thereby reducing the money supply left for transactions in the real sector. But appart from deflation, the economy seems largely unaffected as the produced volumes stay in line with the previous configurations without fundamentalists – at least as long as a minimal amount of money is left.

6.7. Booms and Busts

In order to focus on the stock market dynamics, the previous configuration is simplified. Instead of having a sinoid birth rate, the birth rate is kept constant with a consumer born every ten days. There is only one consumption good 'apples', as well as a few other older settings that apply.[48] The five fundamentalists are not added until day 3000 to better see the impact of their presence. Eventually, the market escalates like in the configuration from the previous section 6.6. Before doing so, it exhibits highly interesting dynamics of booms and busts as the buyback escalation gets interrupted a few times. This configuration can be found in its own branch #FundamentalistBubble, which also serves as data source for the sample web output from appendix A.

Figure 33 charts the average stock price of the fundamentalists, with the correlation between the individual fundamentalists being 0.99. The other stocks follow different, less extreme paths. The apple firms peak whenever the goods prices reach their highs, enabling a high dividend. The shares of the market makers thrive when volatility is highest and when the funds offload their holdings to the market

[48] Life expectancy is 800 days and returns to scale $\lambda = 0.5$. Apple firms still operate under an earlier firm decision heuristic, namely the 'optimal cost' heuristic with $b_R = 1$ presented in section 4.3. There are five market makers and fundamentalists each. The price belief of the market makers is adjusted by a factor of at most 1.5 versus 1.1 in the base case. Furthermore, they offer 10% of their holdings in each ask order instead of just 5%. The reason for all these differences is that this configuration was discovered comparatively early under different defaults. Also, it highly sensitive to parameter adjustments, a defining property of chaotic systems.

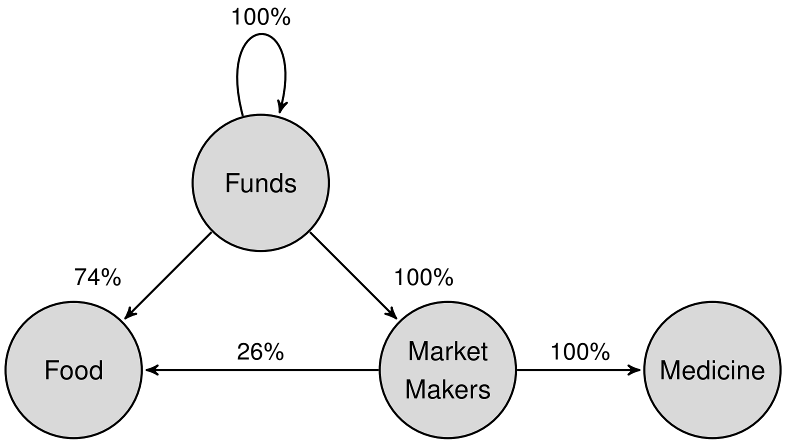


Figure 32: Even though individual agents are not allowed to buy their own shares, funds start to buy each other, leading to the shown ownership structure with funds directly or indirectly owning (almost) everything. Dividends emitted by food and medicine firms end up with the fundamentalist funds, which hoard all money in a circular dividend flow between themselves. This leads to a price deflation in the real sector until – at some point – no cash is left and the economy collapses.

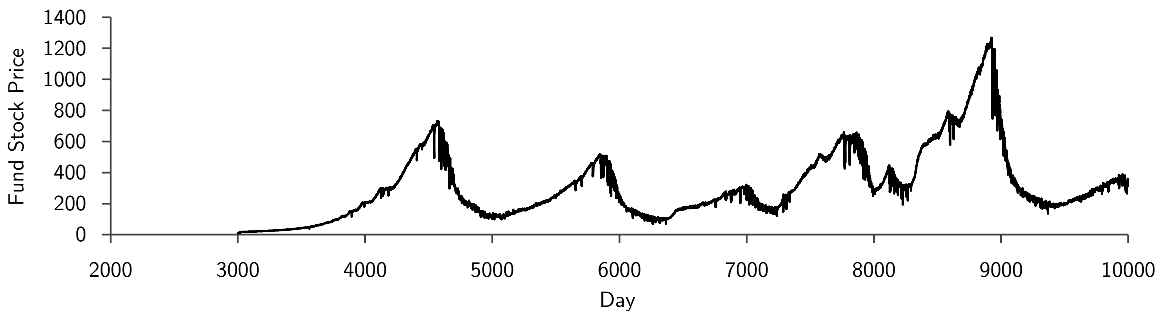


Figure 33: The stock prices of the fundamentalists added on day 3000 follow a recurring boom and bust pattern before they eventually escalate towards infinity as before (not visible here). The buyback escalations are interrupted when their dividend yield temporarily falls below that of all the other companies, triggering a mutual sell-off with the market makers earning good money in the phase of increased volatility following the peak.

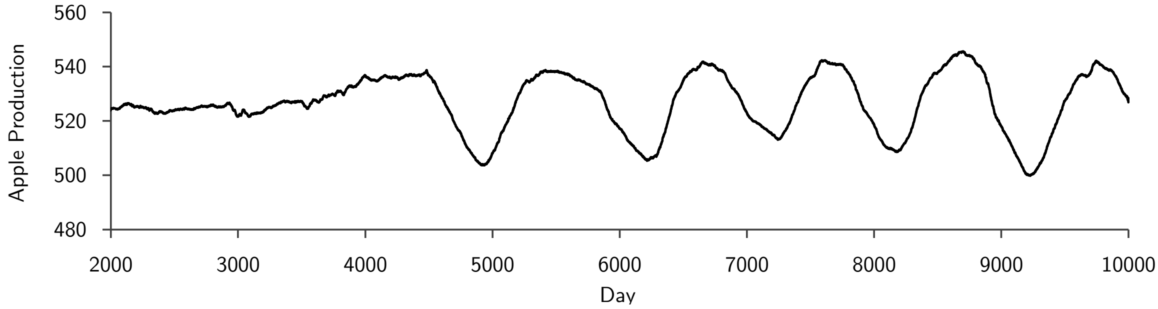


Figure 34: Production volumes mirror the turmoil in the financial markets, with workers periodically working more than optimal or less then optimal, depending on their dividend income.

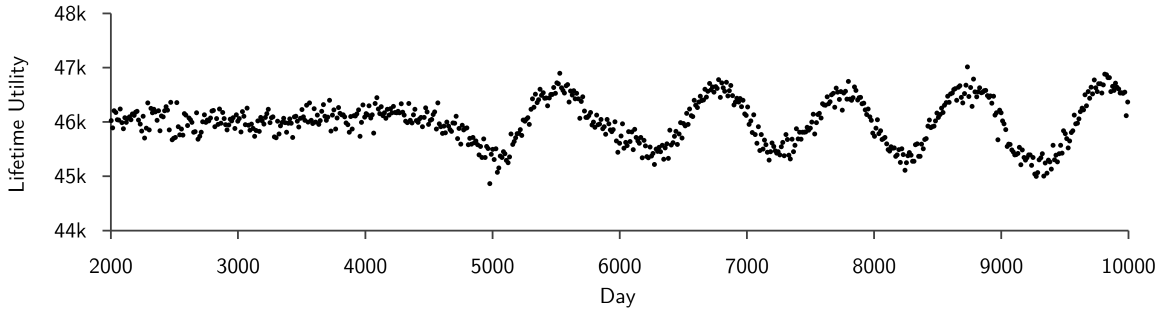


Figure 35: Each dot represents the death of a consumer, with the accumulated life-time utility on the y-axis. Those consumers who can buy shares low when young and sell them high when retired enjoy the highest utility.

makers at a low price.

This time, the turmoil in the financial markets also affects production volumes in the real sector, as shown in figure 34. This demonstrates that the introduction of a trader that follows a seemingly stabilizing value strategy can not only destabilize financial markets, but also distort the underlying real sector. Each individual agent in this simulation behaves according to simple, reasonable heuristics. Yet, an outcome emerges that is not so reasonable any more. The rolling five-day returns of the fundamentalist stocks exhibits a negative skewness of -1.04, and a high excess kurtosis of 22. Both statistics are discussed in section 6.8.

The time of birth determines the total lifetime utility of the individual consumers, shown in figure 35. Consumers are best off if they are born in a period of low stock prices, allowing them to acquire them cheap, and spend their retirements in a period of high prices, allowing them to consume a lot when old. Apparently, it is possible to exploit market turbulences in order to improve one's utility, showing that the simulated stock market is not efficient and that the observed price peaks irrational, indicating that a new way of provoking bubbles in agent-based simulations has been found. Known methods include trend-following chartists, contaminous sentiments, or leveraged trading, as already enumerated earlier in section 6.6.

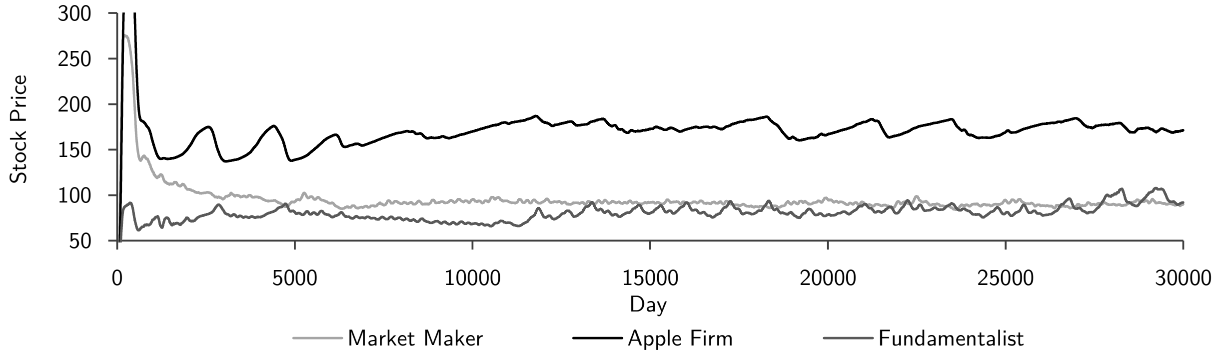


Figure 36: A long-term time series showing the qualitatively different return structures of the three firm types. The initialization spike of the apple farms echoes a few times before its traces disappear. Unlike before, none of the stock prices escalate even when running the simulation for ten million days.

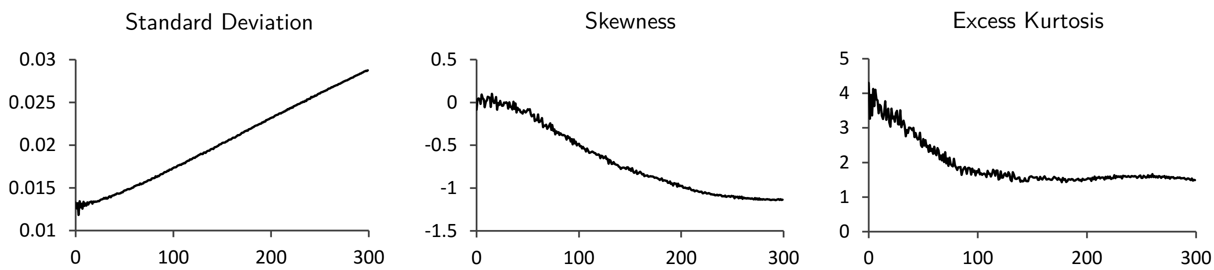


Figure 37: The statistical properties of the log-returns of the average 'apple firm' shares depend on the chosen timeframe s in $log(p_{t+s+1}/p_t)$.

6.8. Skewed Returns

The dynamics of the previous configuration are tamed somewhat in the configuration #Longterm, which is based on the overlapping generations scenario from section 6.4, except that birth rates are flat, that there is only one consumption good 'apples', and that consumers invest in *index* mode. The main difference in comparison to the just discussed #fundamentalistbubble scenario is the presence of twice as many market makers, namely 10, and disallowing them to adjust their price beliefs by more than 10% per day again. This results in the long term chart 36, showing the stock prices of the three firm types. While the returns of the individual firms within a type are highly correlated, they differ between firm types, with the market maker having the flattest curve and the apple firm the most volatile one.

Figure 37 shows how the choice of timeframe affects standard deviation, skewness and excess kurtosis of the apple firm's log returns between day 10'000 and day 30'000. While the standard deviation increases lineraly (at least in the shown range), skewness and kurtosis depend less predictably on the chosen timeframe. As a reference point, Chen et al. (2001) report the skewness of the indices of nine countries to be between -1.84 (SP 500) and -0.15 (CAC 40), based on daily data.

Hommes (2013) notes that "for a given model, it is often hard to pin down what exactly causes certain stylized facts at the macro level in agent-based micro simulations." This also seems the case here, as the previous explanations do not work any more when trying to explain the stock movements of the apple farming. The circular-ownership explanation does not work as these firms are not trading on the stock market. Also, the correlation in equation 10 cannot explain much any more. Without changing

Variable 1	Variable 2	Correlation	Comment
r_{APPL}	v_{APPL}	-0.46	Agents tend to sell faster than they buy.
p_{APPL}	o	0.49	High prices lead to high outflows when sellling,
p_{APPL}	i	0.06	but not to much higher inflows.
r_{APPL}	$i - o$	0.06	Net inflows cannot adequatly explain returns,
r_{FUND}	$i - o$	0.48	except for the fundamentalists' returns.
p_{APPL}	p_{index}	0.28	High consumer prices come with a higher stock price,
p_{index}	d_{APPL}	0.99	and higher dividends its shareholders.
p_{index}	d_{MARK}	0.43	less so for market makers,
p_{index}	d_{FUND}	-0.24	and not at all for fundamentalists.
p_{APPL}	$own_{F \to A}$	0.25	Funds owning many farm shares comes with higher prices.

Table 6: Variables that correlate with the stock price p_{APPL} of the apple firm or its log return r_{APPL}. All correlations are based on moving averages over 100 days.

population structure, it falls to 0.1. Table 6 lists correlations of interest.[49]

Whereas relations between variables can be shown mathematically in equation-based models, one often needs to resort to econometrics when analyzing agent-based models with chaotic properties. In this regard, agent-based models are closer to reality, but not in a helpful way.

When attempting to explain the negative skewness, the negative correlation between returns and volume might be a starting point. The apple farm's skewness could be caused by agents selling fast and buying slowly, causing a slow upwards movements and fast downwards movements. However, it is not clear which agent should be responsible.

The consumers work for longer than they are retired, and thus individually buy slower than they sell. But in aggregate, this should not make a difference as there are also more workers than retirees at any given point in time. This hypothesis can also be falsified by simply adjusting the retirement age to 500 days and running the simulation again (configuration #LongtermWithRetirementAt500).

Another hypothesis is that the skewness might be caused by the asymmetry between bids and asks of the market maker. By default, the market makers offer 5% of their holdings on the market every day. By adjusting that number and observing the impact on the metrics of interest, one should be able to quickly verify this hypothesis. Except that this is a chaotic system and a linear change in a parameter can cause non-linear responses. In fact, adjusting this parameter to 4% brings back the buyback escalation as shown in figure 38, whereas increasing it to 6% reveals an entirely different, cyclical pattern for the apple firm stock as shown in figure 39. Under such circumstances, the search for linear dependencies is largely pointless.

What can be said for certain, however, is that the root cause of all observed chaotic behavior lies in circular ownership structures between the fundamentalists. Disallowing them from buying each others shares makes the escalations disappear, but the other interesting dynamics vanish with them.

[49] A large spreadsheet with much more data and correlations is available from github.com/kronrod/agentecon/blob/LongtermData/data/longterm.ods.

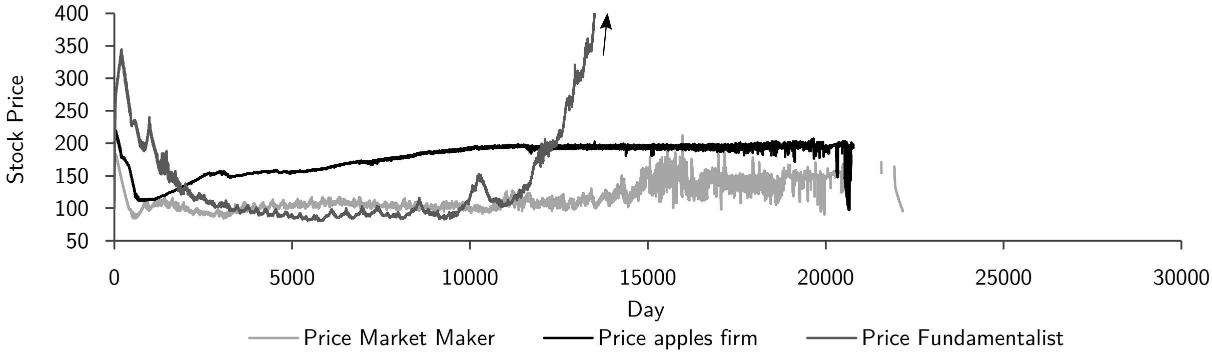


Figure 38: By default, the market maker offers 5% of its portfolio for sale every day. Adjusting this fraction to 4% brings back the previous price escalation with a subsequent collapse of the economy as it runs out of cash. (In this case, the initial price beliefs of the market makers have been adjusted to 100$ instead of 10$ for each stock, delaying the escalation a little and resulting in a nicer chart. Source tag: #Longterm004-100)

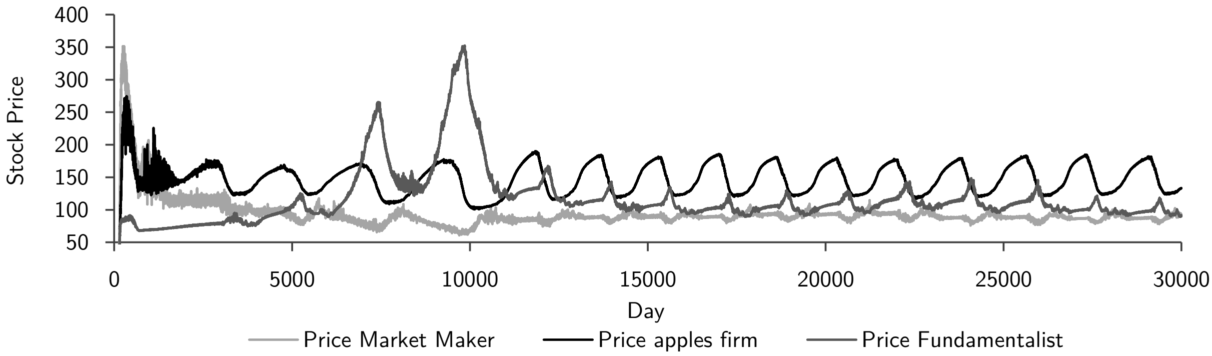


Figure 39: By default, the market maker offers 5% of its portfolio for sale every day. Adjusting this fraction to 6% lets the simulation stabilize on a completely different, cyclical pattern. Source tag: #Longterm006

6.9. Reflection

At some point, it is probably better to admit to be lost in the "wilderness of disequilibrium economics"[50] and take a step back instead of further venturing forward. The constructed stock market started to show chaotic behavior with the introduction of the fundamentalist funds. This in itself is remarkable as these fundamentalists follow seemingly innocuous rules. While plausible explanations for some of the emergent dynamics have been identified, the dynamics that cause the skewness remain unclear. Future research might try to gain a better understanding of the rules that govern the individual agents, maybe similar to the analysis of the firm decision heuristics in section 4.3. The statistical resemblance of some of the observed time series to real-world stock markets is interesting, but not worth much without a better quantitative or other benchmark. In fact, resorting to the verification of stylized facts such as skewed returns is what most authors do when evaluating more elaborate agent-based models, and exactly what was criticized in section 2.5.

[50] The term "wilderness of disequilibrium economics" coined by Sims (1980) has struck a chord among authors of agent-based models and can be found in various variants, but most often as the "wilderness of bounded rationality".

7. Conclusion

Agent-based modeling not only allows to build models from the bottom up, it requires one to do so. Exogenously imposing a desired aggregate outcome – for example a certain price at a certain point in time – is generally impossible. This forces the architect of the model to thoroughly analyze the micro-mechanics of the markets and to carefully design the behavior of each agent with the potential consequences for the aggregate outcome in mind. And even then, the outcomes can be chaotic and hard to predict. One could argue that this makes agent-based models more realistic, but that is not helpful. Instead, well-defined and clear benchmarks are needed to serve as a compass when navigating this wilderness. Without them, the researcher is lost and even the most interesting results are detached islands of knowledge without practical use. The book *A New Kind of Science* by Stephen Wolfram (2002) provides a prime example. It is a brilliant work by a brilliant scientist about a topic not entirely unrelated to agent-based modeling, but it did not live up to its promise so far for being too detached from existing research. In this thesis, I followed the more fruitful and pragmatist approach of replicating proven theory before daring to undertake a modest exploration into the wilderness. While the latter is more entertaining, the former produced more tangible results in the form of a suite of complementary tools and methods that hopefully prove helpful in constructing future agent-based models.

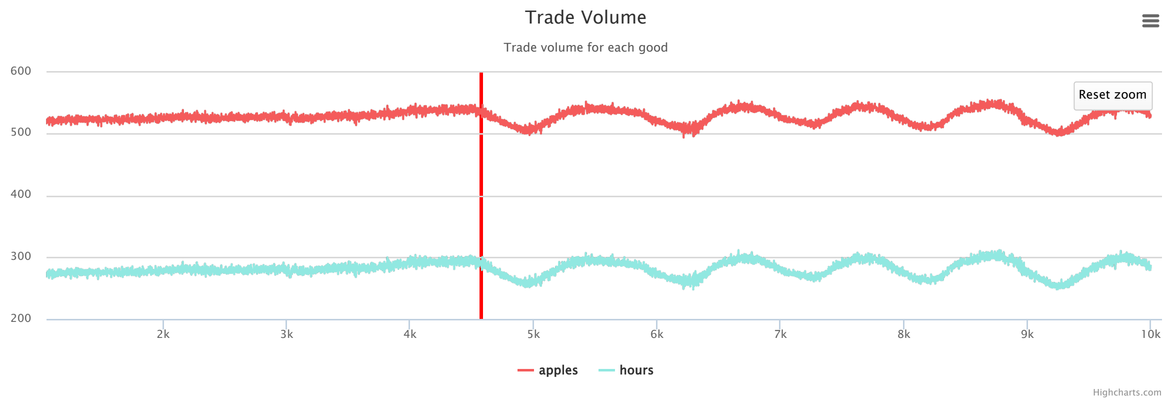
A. Web Visualization

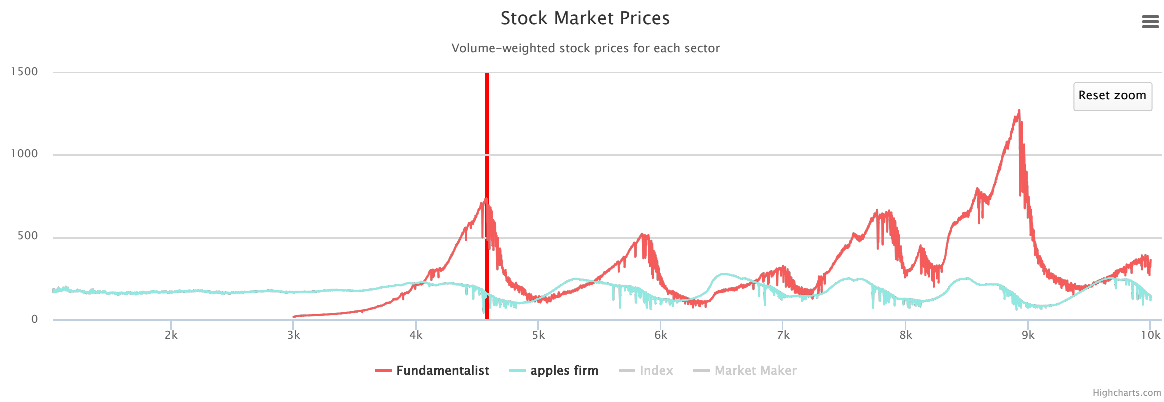
A printout of the website master.agentecon.com/sim.html?id=FundamentalistBubble, visualizing the results of running #FundamentalistBubble. On the website, the red line can be positioned by clicking into any chart, it always marks the same day in all of them. The charts can all be zoomed into and each time series toggled on and off. For further information, see section 2.6.

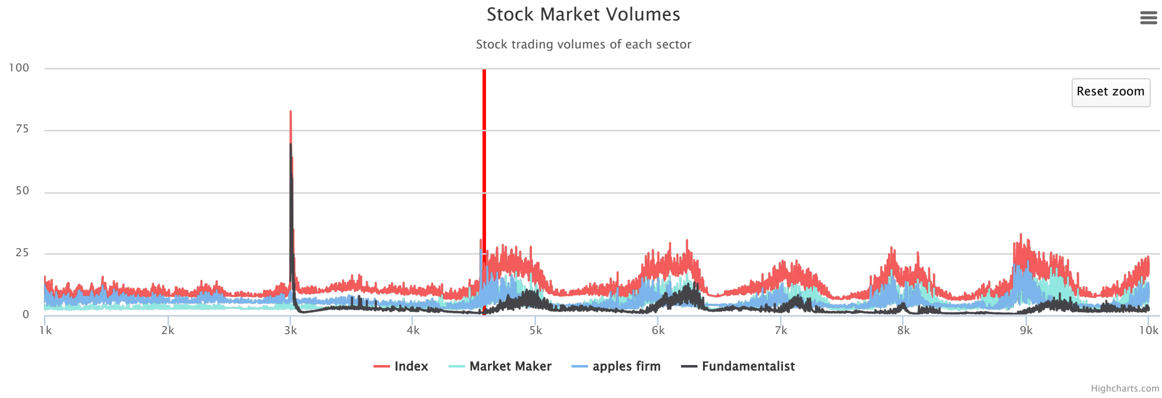
FundamentalistBubble

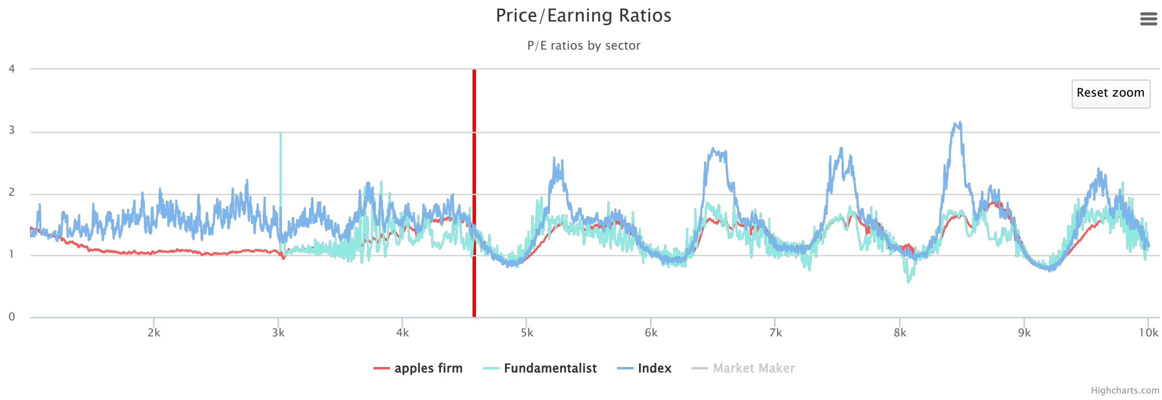
Commit comment: added support for inflow vs outflow chart

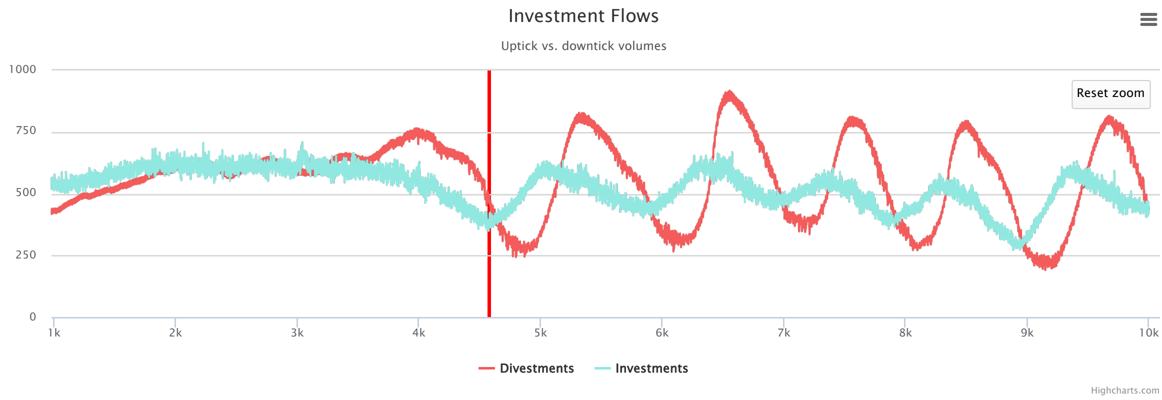
Click on legend items to filter them. Zoom into the charts by selecting the area of interest. Once zoomed in, hold 'shift' to pad. Semi-transparent areas span the range between minimum and maximum observed value, for example in the price chart. Charts might be slightly inaccurate as some data points are skipped in order to ensure acceptable browser performance.
Feel free to browse this simulation's source code (https://github.com/kronrod/agentecon/tree/FundamentalistBubble) on github or download and run it yourself.

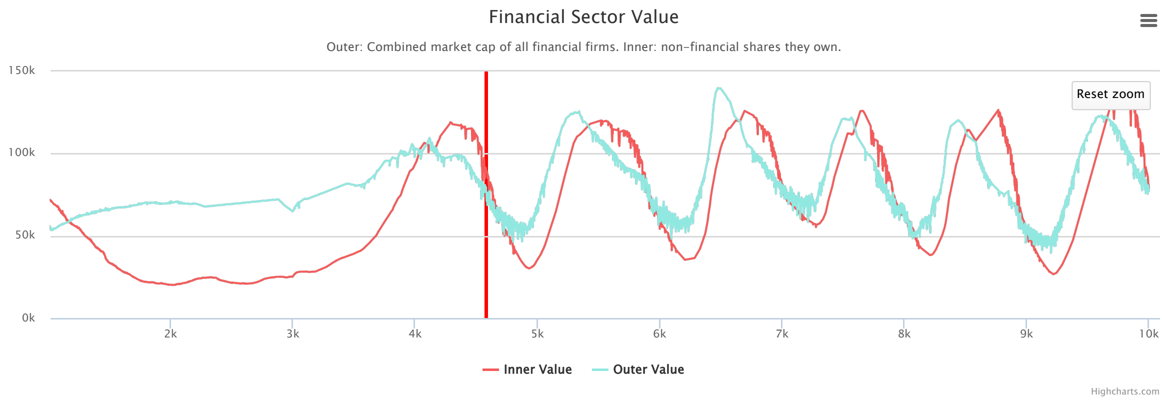


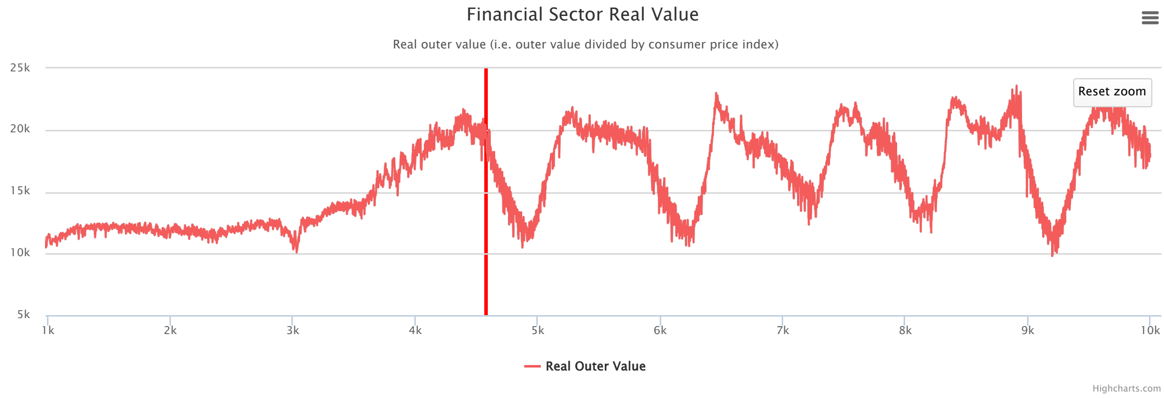


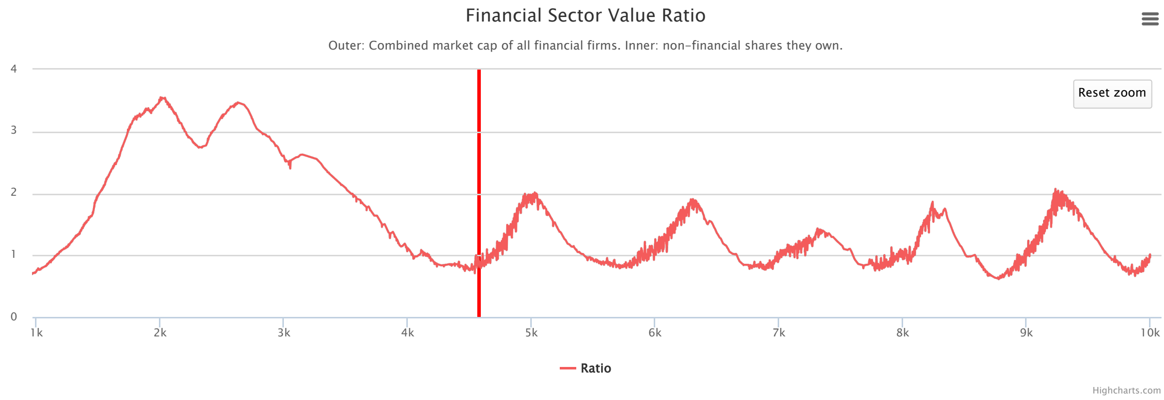

Stock Market Volumes
Stock trading volumes of each sector
Reset zoom
100
75
50
25
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Index
Market Maker
apples firm
Fundamentalist
Highcharts.com

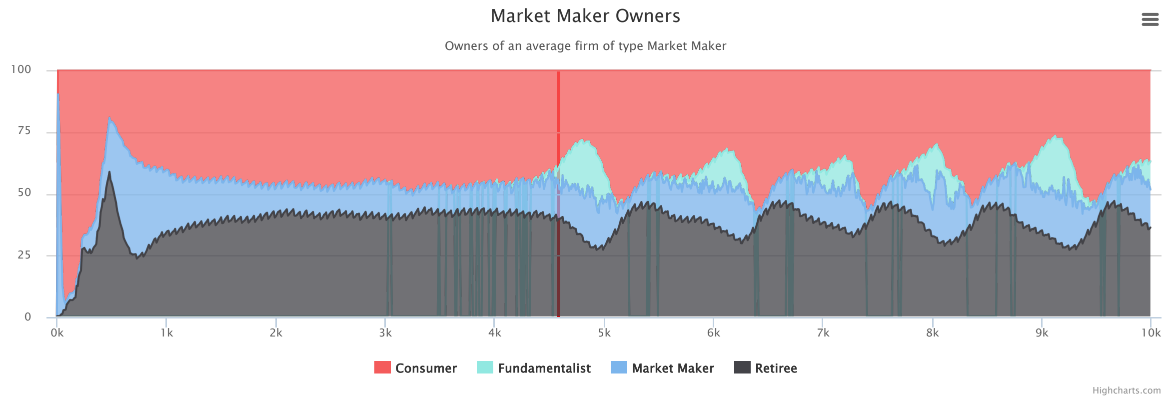

Price/Earning Ratios
P/E ratios by sector
Reset zoom
4
3
2
1
0
2k
3k
4k
5k
6k
7k
8k
9k
10k
apples firm
Fundamentalist
Index
Market Maker
Highcharts.com

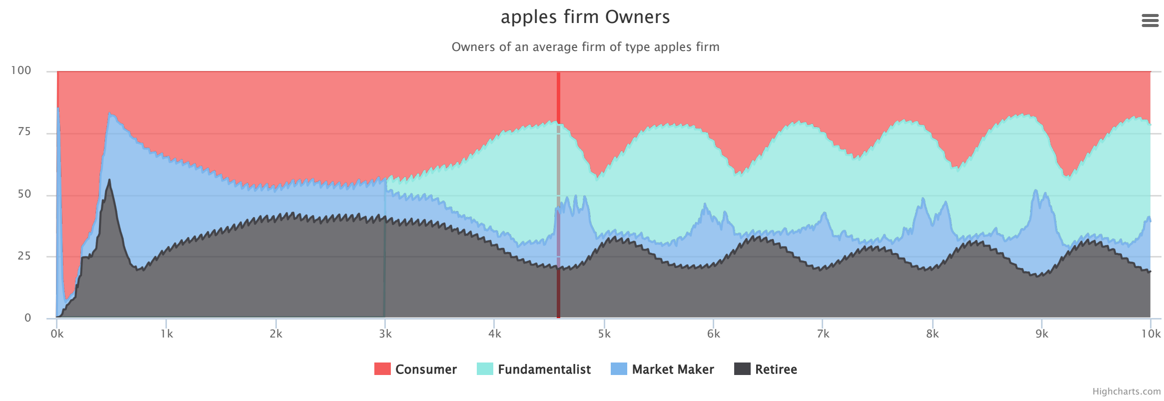

Investment Flows
Uptick vs. downtick volumes
Reset zoom
1000
750
500
250
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Divestments
Investments
Highcharts.com

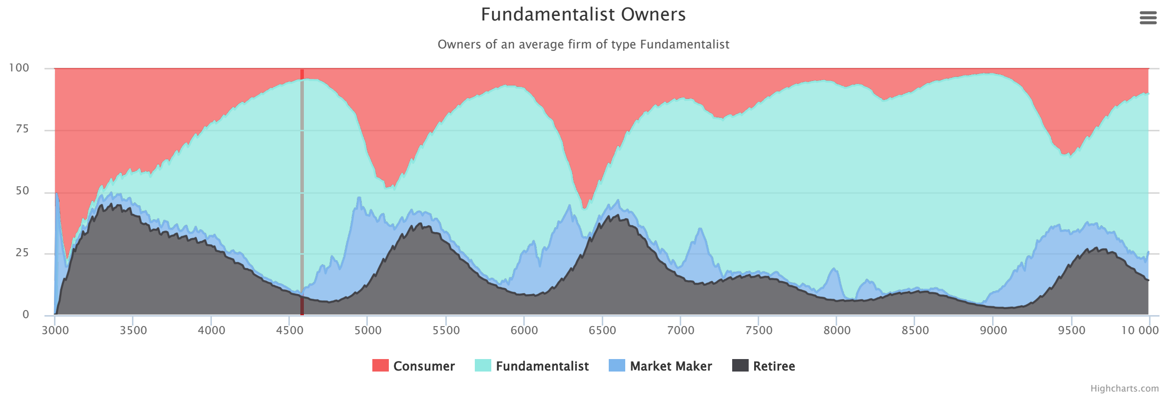

Financial Sector Value
Outer: Combined market cap of all financial firms. Inner: non-financial shares they own.
Reset zoom
150k
100k
50k
0k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Inner Value
Outer Value
Highcharts.com

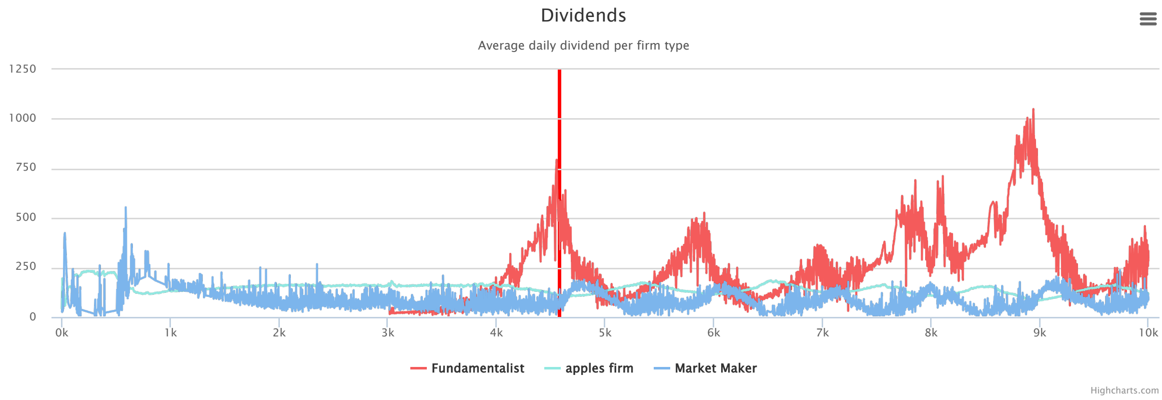

Financial Sector Real Value
Real outer value (i.e. outer value divided by consumer price index)
Reset zoom
25k
20k
15k
10k
5k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Real Outer Value
Highcharts.com

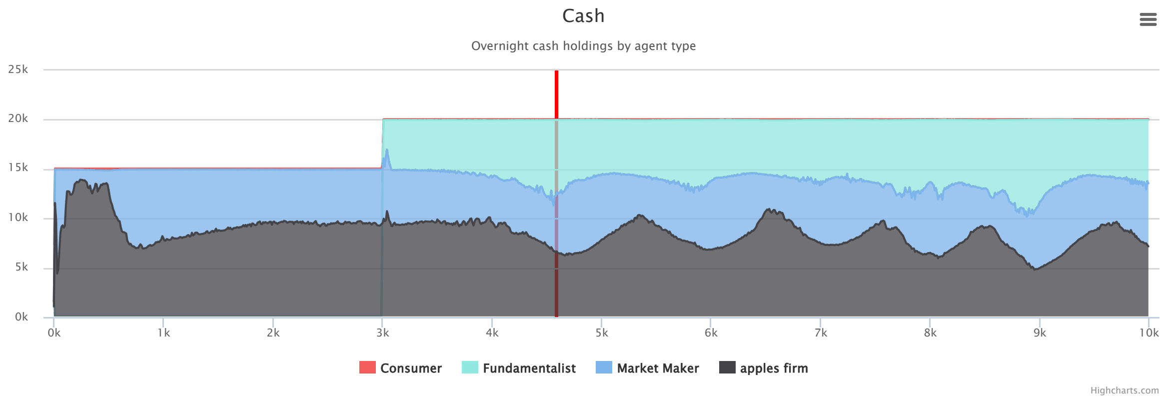

Financial Sector Value Ratio
Outer: Combined market cap of all financial firms. Inner: non-financial shares they own.
Reset zoom
4
3
2
1
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Ratio
Highcharts.com

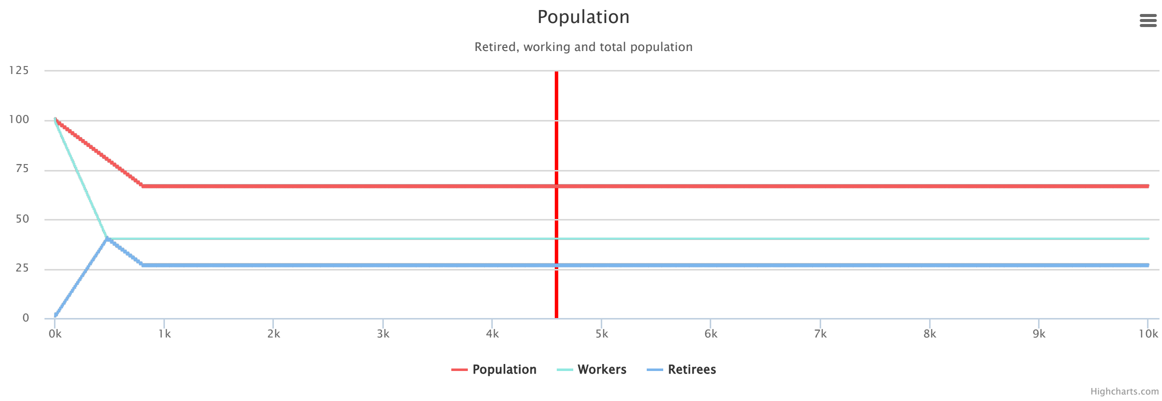

Market Maker Owners
Owners of an average firm of type Market Maker
100
75
50
25
0
0k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Consumer
Fundamentalist
Market Maker
Retiree
Highcharts.com

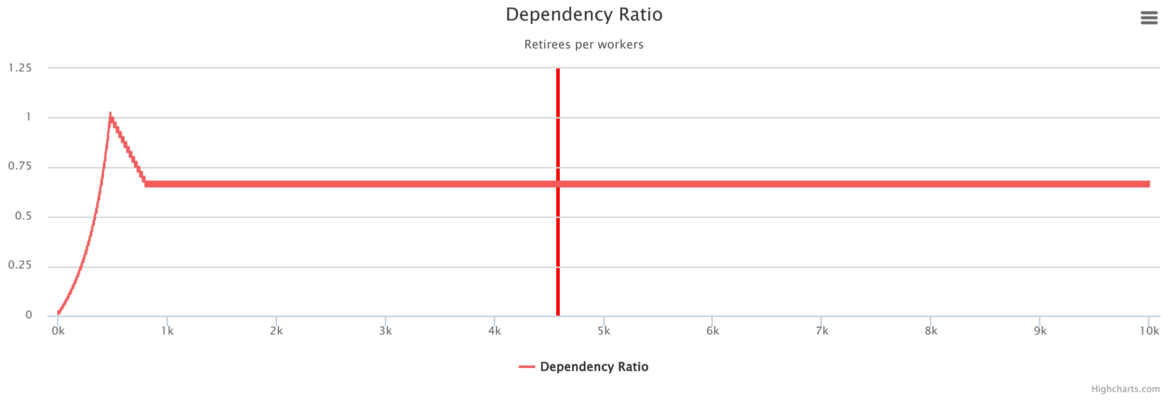

apples firm Owners
Owners of an average firm of type apples firm
100
75
50
25
0
0k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Consumer
Fundamentalist
Market Maker
Retiree
Highcharts.com

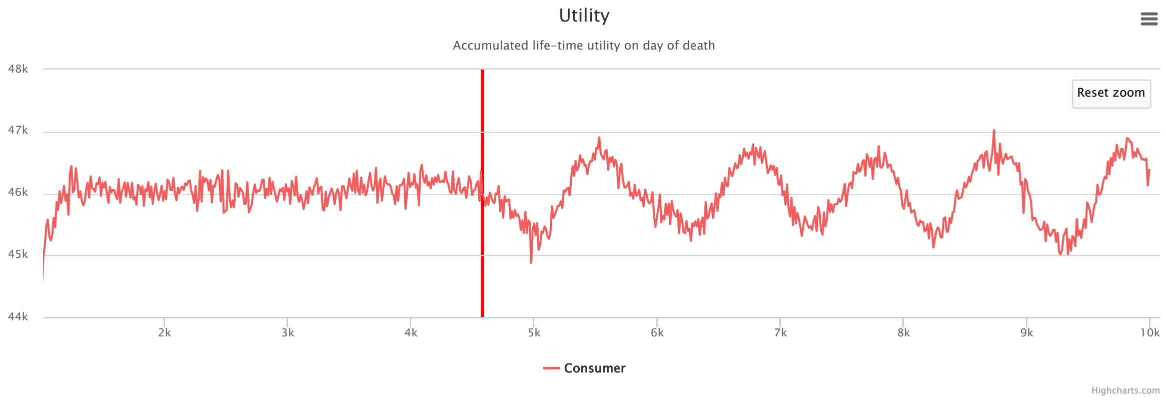

Fundamentalist Owners
Owners of an average firm of type Fundamentalist
100
75
50
25
0
3000
3500
4000
4500
5000
5500
6000
6500
7000
7500
8000
8500
9000
9500
10 000
Consumer
Fundamentalist
Market Maker
Retiree
Highcharts.com


Dividends
Average daily dividend per firm type
1250
1000
750
500
250
0
0k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Fundamentalist
apples firm
Market Maker
Highcharts.com


Cash
Overnight cash holdings by agent type
25k
20k
15k
10k
5k
0k
0k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Consumer
Fundamentalist
Market Maker
apples firm
Highcharts.com


Population
Retired, working and total population
125
100
75
50
25
0
0k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Population
Workers
Retirees
Highcharts.com


Dependency Ratio
Retirees per workers
1.25
1
0.75
0.5
0.25
0
0k
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Dependency Ratio
Highcharts.com


Utility
Accumulated life-time utility on day of death
Reset zoom
48k
47k
46k
45k
44k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Consumer
Highcharts.com

```
Correlation List
Corr(Population, Dependency Ratio)	1
Corr(Retirees, Dependency Ratio)	1
Corr(Retirees, Population)	1
Corr(Price apples, Price Index) 0.999
Corr(Dividends apples firm, Overnight cash apples firm) 0.998
Corr(Price hours, Price Index)	0.996
Corr(Price hours, Price apples) 0.99
Corr(Price apples, Dividends apples firm)	0.982
Corr(Price apples, Overnight cash apples firm)	0.981
Corr(Price Index, Dividends apples firm)	0.978
Corr(Dividends Fundamentalist, Overnight cash Fundamentalist)	0.976
Corr(Price Index, Overnight cash apples firm)	0.975
Corr(Price hours, Dividends apples firm)	0.963
Corr(Price hours, Overnight cash apples firm)	0.957
Corr(P/E Ratio Market Maker, P/E Ratio Index)	0.953
Corr(Price apples firm, Divestments)	0.934
Corr(Price Fundamentalist, Dividends Fundamentalist)	0.931
Corr(Price Fundamentalist, Overnight cash Fundamentalist)	0.926
Corr(Volume hours, Volume apples)	0.925
Corr(P/E Ratio apples firm, Inner Value)	0.912
Corr(Divestments, Overnight cash Consumer)	0.891
Corr(Inner Value, Real Outer Value)	0.89
Corr(Price apples firm, Outer Value)	0.883
Corr(Price hours, Investments)	0.878
Corr(apples/Price Index, Volume apples) 0.86
Corr(P/E Ratio apples firm, Real Outer Value)	0.848
Corr(Price Index, Investments)	0.841
Corr(Price apples firm, P/E Ratio Index)	0.836
Corr(Divestments, Outer Value)	0.828
Corr(P/E Ratio Index, Divestments)	0.827
Corr(Price apples firm, Overnight cash Consumer)	0.824
Corr(Price apples, Investments) 0.819
Corr(Divestments, Overnight cash apples firm)	0.812
Corr(Volume apples, P/E Ratio apples firm)	0.811
Corr(Volume hours, P/E Ratio apples firm)	0.809
Corr(Price Index, Price apples firm)	0.809
Corr(Divestments, Dividends apples firm)	0.805
Corr(Outer Value, Real Outer Value)	0.805
Corr(apples/Price Index, Volume hours)	0.804
Corr(Volume hours, Utility on death Consumer)	0.804
Corr(P/E Ratio Fundamentalist, Outer Value)	0.79
Corr(Volume apples, Utility on death Consumer)	0.789
Correlation List
Corr(hours/Price Index, apples/Price Index)	-0.999
Corr(Divestments, Overnight cash Market Maker)	-0.889
Corr(Investments, Overnight cash Fundamentalist)	-0.863
Corr(hours/Price Index, Volume apples)	-0.852
Corr(Investments, Dividends Fundamentalist)	-0.851
Corr(Price apples firm, Overnight cash Market Maker)	-0.838
Corr(Price Fundamentalist, Investments) -0.818
Corr(Outer Value, Overnight cash Market Maker)	-0.809
Corr(hours/Price Index, Volume hours)	-0.796
Corr(P/E Ratio Index, Overnight cash Market Maker)	-0.784
Corr(Overnight cash Market Maker, Overnight cash Consumer)	-0.781
Corr(Price hours, Overnight cash Fundamentalist)	-0.779
Corr(Price hours, Dividends Fundamentalist)	-0.77
Corr(hours/Price Index, P/E Ratio apples firm)	-0.751
Corr(Volume apples, Overnight cash Market Maker)	-0.749
Corr(Price Index, Overnight cash Fundamentalist)	-0.746
Corr(Volume hours, Overnight cash Market Maker) -0.743
Corr(hours/Price Index, Utility on death Consumer)	-0.743
Corr(Price Index, Dividends Fundamentalist)	-0.736
Corr(Price apples, Overnight cash Fundamentalist)	-0.727
Corr(P/E Ratio Market Maker, Overnight cash Market Maker)	-0.722
Corr(Volume Market Maker, Overnight cash apples firm)	-0.717
Corr(Price apples, Dividends Fundamentalist)	-0.717
Corr(Volume Market Maker, Dividends apples firm)	-0.715
Corr(apples/Price Index, Overnight cash Market Maker)	-0.708
Corr(hours/Price Index, Divestments)	-0.707
Corr(Price apples, Volume Market Maker) -0.693
Corr(Price Index, Overnight cash Market Maker)	-0.692
Corr(hours/Price Index, Price apples firm)	-0.688
Corr(Price Index, Volume Index) -0.688
Corr(Price Index, Volume Market Maker)	-0.684
Corr(hours/Price Index, Inner Value)	-0.68
Corr(Volume Market Maker, Divestments)	-0.673
Corr(P/E Ratio Index, Dividends Market Maker)	-0.667
Corr(Dividends apples firm, Overnight cash Fundamentalist)	-0.663
Corr(Price hours, Volume Market Maker)	-0.662
Corr(Price hours, Price Fundamentalist) -0.66
Corr(Overnight cash apples firm, Overnight cash Fundamentalist) -0.658
Corr(Dividends apples firm, Dividends Fundamentalist)	-0.65
Corr(P/E Ratio apples firm, Overnight cash Market Maker)	-0.647
Corr(Dividends Fundamentalist, Overnight cash apples firm)	-0.646
Corr(Overnight cash Market Maker, Overnight cash apples firm)	-0.639
```

B. Software Architecture

This appendix provides an overview over the software architecture at a high level and motivates a few of the design decisions. It is a good starting point for readers that wish to replicate the presented results themselves or want to find inspirations for implementing similar models.

B.1. Cloud Execution

As shown in figure 3, the simulation can be executed in the cloud with its result visible in the web. This component is built on Google App Engine, which was chosen for the high level of abstraction it offers and the good integration with the Eclipse source code editor, from where updates can be pushed to the web with a single click. It is split into two projects, a frontend AgenteconFront and a backend AgenteconBack.[51]

The frontend serves the website and loads simulation metadata such as tags and commit comments directly from GitHub through the offered API. When displaying a simulation, it tries to load the according charts from the App Engine Datastore. If there is none, it adds a request to run that simulation to a task queue. Both the datastore and the task queue are managed and run by Google, so there is no need to manually setup anything or to bother with backups.

The backend processes the simulation requests from the queue, and stores the results in the datastore. It also stores intermediate results, but the user needs to actively refresh the browser while the simulation is running in order to see them. App engine can be configured to use any number of instances to process the task queue, but it only offers nine standard instance-hours on "B2" instances for free per day. Running four simulations in parallel on better "B8" instances with 1GB of ram and 4.8GHz already depletes that free quota within half an hour. Unfortunately, app engine turned out to be about ten times slower at executing the simulations than my own PC with the same nominal speed. This turned out to be caused by Google monitoring every function call in order to ensure that the instances cannot see data of other app engine users.[52]

Such a cloud setup has various advantages. It allows to publicly link to simulation results (unfortunately, the website is not stable and tested enough across different platforms to confidently do so) without cumbersome downloads. Furthermore, it is possible to run hundreds simulations in parallel when desired with minimal effort in comparison to the typical computing clusters for scientific computing, which require formalities to gain access to and where parallelization must be dealt with much more explicitely by the programmer. Unfortunately, doing so in App Engine could turn out to be quite costly, as each instance-hour above the free quota costs 0.64$. Given its abismal performance, running CPU-bound tasks on app engine is not cost-effective. Thus, I cannot recommend using it for scientific computing as long as there are no significant improvements.

The cloud frontend and backend together consist of 49 classes and about 2800 lines of code, including tests but excluding html and javascript files.

[51] The source code is publicly available from bitbucket.org/Kronrod/agentecon.

[52] Apparently, the responsible engineers at Google chose a design that allows many instances of different users to run within the same Java Virtual Machine, enabling faster startup and reducing the memory-footprint, but necessitating security checks they implemented in a computationally costly way. When running simulations in the local test version of App Engine, these checks can be disabled using the undocumented vm argument *-Dappengine.disableRestrictedCheck="true"*, which I found out about through reverse engineering their eclipse plugin.

B.2. Interface

The cloud backend must be able to run many versions of the same simulation. This is only possible if the different versions of the same classes and functions can be cleanly separated from each other in memory despite their identical names. In Java, this is usually done by instantiating a separate classloader for each simulation run. These classloaders can dynamically load code from arbitrary sources – in this case from the github repository where the different versions of the simulations reside. However, since one can generally not know in advance what these simulations look like and what classes they consist of, it is necessary to specify a specific simulation interface that all versions must adhere to.[53]

The interface contains the necessary methods to run a simulation and to extract data from it in order to record time series or other statistical information. It also provides some of the basic classes used in the simulations, such as inventories of goods and portfolios of shares, which allows the simulation run to easily extract data of interest from these classes.

The interface consists of 39 classes with 1200 lines of code, including tests.

B.3. Simulation

The architecture of the simulation closely follows the composition of the agents as they are presented in the other chapters, with each agent being represented in its own class, with functionality added to those classes by use of composition[54] – i.e. by having a common portfolio class used by all agent types that trade on the stock market. Furthermore, many synchronization pitfalls can be avoided by strictly enforcing hierarchical control, i.e. never allowing agents to invoke methods on each other. For example, firms do not send dividends to its shareholders directly. Instead, they deposit the dividends with each position in the shareholder register, from where the shareholders later transfer the dividends into their own wallet. This approach allows to rule out circular dependencies, which can often lead to subtle errors.

The source code of all simulations is hosted on github.com/kronrod/agentecon, with source tags used to directly refer each discussed configuration (see also section 2.6). The latest version of the simulation consists of 146 classes and 7500 lines of code including tests.

[53] Available for download as compiled jar-file including source code from github.com/kronrod/agentecon/blob/master/jar/agenteconinterface.jar.

[54] When in doubt, composition should be preferred over inheritance. The fact that inheritance is a unique feature of object-oriented languages leads many to wrongly believe that object-oriented programs must make extensive use of it. In practice, however, it is a clean encapsulation that makes an object-oriented design good, regardless of whether inheritance is used or not.

C. Revised Version of *An Agent-Based Simulation of the Stolper-Samuelson Effect*

This appendix contains the completely revised version of *An Agent-Based Simulation of the Stolper-Samuelson Effect* submitted to *Computational Economics* for inclusion in a special CEF 2015 issue, the conference where the earlier version was presented (Computating in Economics and Finance, June 2015, Taipei, www.aiecon.org/conference/cef2015). The acceptance decision is planned for April 2016, with an initial review by end of 2015.

The earlier version consisted of a minimal agent-based model capable of reproducing the Stolper-Samuelson effect. It already made use of exponential search, but not of the other presented techniques, and was also limited in other regards, for example not having profit-maximizing firms. Further information can be found in section 1.

Declaration of Authorhips: Friedrich Kreuser is a co-author of both papers. However, all content that appears in both, this thesis and the revised paper, has been written by me as part of this thesis, with Friedrich proofreading and generating figures 8 to 11 from provided data. Also, some of the source code from the earlier paper has been reused in this thesis. This earlier code can be found on github.com/kronrod/ace-stolper-samuelson.

In the print version, the first revision of paper *An Agent-Based Simulation of the Stolper-Samuelson Effect* is attached at this point. Here, in the online version of this thesis, a link is provided instead: master.agentecon.com/draft.pdf.

References

J. Arifovic. Evolutionary algorithms in macroeconomic models. *Macroeconomic Dynamics*, 4(03):373–414, 2000.

K. J. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.

S. B. Aruoba and J. Fernández-Villaverde. A comparison of programming languages in economics. Technical report, National Bureau of Economic Research, 2014.

R. Axtell. *The emergence of firms in a population of agents: local increasing returns, unstable Nash equilibria, and power law size distributions*. Center on Social and Economic Dynamics Brookings Institution Discussion paper, 1999.

R. Axtell. Zipf distribution of us firm sizes. *Science*, 293(5536):1818–1820, 2001.

R. Axtell. The complexity of exchange. *The Economic Journal*, 115(504):F193–F210, 2005.

J. L. Bentley and A. C.-C. Yao. An almost optimal algorithm for unbounded searching. *Information processing letters*, 5(3):82–87, 1976.

T. Brenner. Agent learning representation: advice on modelling economic learning. *Handbook of computational economics*, 2:895–947, 2006.

W. A. Brock and C. Hommes. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic dynamics and Control*, 22(8):1235–1274, 1998.

R. Bruno. Version control systems to facilitate research collaboration in economics. *Computational Economics*, pages 1–7, 2015.

J. Bullard and J. Duffy. Using genetic algorithms to model the evolution of heterogeneous beliefs. *Computational Economics*, 13(1):41–60, 1999.

A. Caiani, A. Godin, S. Kinsella, E. Caverzas, A. Russo, L. Riccetti, J. Stiglitz, and M. Gallegati. Innovation, demand, and finance in an agent based-stock flow consistent model. In *Social Simulation Conference*, 2014.

A. Caiani, A. Godin, E. Caverzasi, M. Gallegati, S. Kinsella, and J. E. Stiglitz. Agent based-stock flow consistent macroeconomics: Towards a benchmark model. *Available at SSRN*, 2015.

M. Catalano and C. Di Guilmi. Dynamic stochastic generalised aggregation in a multisectoral macroeconomic model. In *Proc. of the 21st Int. Conf. on Computing in Economics and Finance (CEF 2015)*, 2015.

A. C. Chang and L. Phillip. Is economics research replicable? sixty published papers from thirteen journals say 'usually not'. *Finance and Economics Discussion Series*, 083, 2015.

G.-m. Chen, M. Firth, and O. M. Rui. The dynamic relation between stock returns, trading volume, and volatility. *Financial Review*, 36(3):153–174, 2001.

I.-K. Cho and T. J. Sargent. Self-confirming equilibrium. *The new Palgrave dictionary of economics*, 2008.

S. Cincotti, D. Sornette, P. Treleaven, S. Battiston, G. Caldarelli, C. Hommes, and A. Kirman. An economic and financial exploratory. *The European Physical Journal Special Topics*, 214(1):361–400, 2012.

R. H. Coase. The nature of the firm. *Economica*, 4(16):386–405, 1937.

O.-J. Dahl and K. Nygaard. Simula: an algol-based simulation language. *Communications of the ACM*, 9 (9):671–678, 1966.

A. Deaton and S. Zaidi. *Guidelines for constructing consumption aggregates for welfare analysis*, volume 135. World Bank Publications, 2002.

C. Deissenberg, S. Van Der Hoog, and H. Dawid. Eurace: A massively parallel agent-based model of the european economy. *Applied Mathematics and Computation*, 204(2):541–552, 2008.

J. Duffy. Agent-based models and human subject experiments. *Handbook of computational economics*, 2: 949–1011, 2006.

W. Elmaghraby and P. Keskinocak. Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science*, 49(10):1287–1309, 2003.

J. M. Epstein. *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, 2006.

K. Etessami and M. Yannakakis. On the complexity of nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.

J. D. Farmer and D. Foley. The economy needs agent-based modelling. *Nature*, 460(7256):685–686, 2009.

J. D. Farmer, M. Gallegati, C. Hommes, A. Kirman, P. Ormerod, S. Cincotti, A. Sanchez, and D. Helbing. A complex systems approach to constructing better models for managing financial markets and the economy. *The European Physical Journal Special Topics*, 214(1):295–324, 2012.

A. M. Feldman. Bilateral trading processes, pairwise optimality, and pareto optimality. *The Review of Economic Studies*, pages 463–473, 1973.

I. Fisher. *The Purchasing Power of Money*. Library of Economics and Liberty, 1922.

D. B. Fogel. *Blondie24: Playing at the Edge of AI*. Morgan Kaufmann, 2001.

J. A. Frankel and K. A. Froot. Chartists, fundamentalists, and trading in the foreign exchange market. *The American Economic Review*, pages 181–185, 1990.

D. Fudenberg and D. K. Levine. Self-confirming equilibrium. *Econometrica: Journal of the Econometric Society*, pages 523–545, 1993.

D. D. Gatti, S. Desiderio, E. Gaffeo, P. Cirillo, and M. Gallegati. *Macroeconomics from the Bottom-up*, volume 1. Springer Science & Business Media, 2011.

J. Geanakoplos, M. Magill, and M. Quinzii. Demography and the long-run predictability of the stock market. *Brookings Papers on Economic Activity*, 2004(1):241–325, 2004.

S. Ghosal and J. Porter. Decentralised exchange, out-of-equilibrium dynamics and convergence to efficiency. *Mathematical Social Sciences*, 66(1):1–21, 2013.

H. Gintis. The dynamics of general equilibrium. *The Economic Journal*, 117(523):1280–1309, 2007.

S. M. Goldman and R. M. Starr. Pairwise, t-wise, and pareto optimalities. *Econometrica: Journal of the Econometric Society*, pages 593–606, 1982.

R. M. Goodwin. Iteration, automatic computers, and economic dynamics. *Metroeconomica*, 3(1):1–7, 1951.

V. Grimm, E. Revilla, U. Berger, F. Jeltsch, W. M. Mooij, S. F. Railsback, H.-H. Thulke, J. Weiner, T. Wiegand, and D. L. DeAngelis. Pattern-oriented modeling of agent-based complex systems: lessons from ecology. *Science*, 310(5750):987–991, 2005.

V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, et al. A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1):115–126, 2006.

V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback. The odd protocol: a review and first update. *Ecological modelling*, 221(23):2760–2768, 2010.

J. Gutknecht and J. Hromkovic. Plädoyer für den programmier-unterricht. *Mitteilungen der Schweizer Informatik Gesellschaft*, 33(2):230–238, 2010. ISSN 0170-6012. doi: 10.1007/s00287-010-0432-8.

F. A. Hayek. The use of knowledge in society. *The American Economic Review*, 35(4), 1945.

T. Hens and B. Pilgrim. *General equilibrium foundations of finance: structure of incomplete markets models*, volume 33. Springer Science & Business Media, 2002.

C. Hommes. *Behavioral rationality and heterogeneous expectations in complex economic systems*. Cambridge University Press, 2013.

C. Hommes and D. in 't Veld. Booms, busts and behavioural heterogeneity in stock prices. *Tinbergen Institute Discussion Paper*, 2014.

D. H. Kim. Guidelines for drawing causal loop diagrams. *The Systems Thinker*, 3(1):5–6, 1992.

E. Landis, I. Yaglom, V. Brudno, W. Gautschi, and M. Senechal. Remembering as kronrod. *The Mathematical Intelligencer*, 24(1):22–30, 2002.

B. LeBaron. Empirical regularities from interacting long-and short-memory investors in an agent-based stock market. *Evolutionary Computation, IEEE Transactions on*, 5(5):442–455, 2001.

B. Lebaron. Building the santa fe artificial stock market. In *School of International Economics and Finance, Brandeis*, pages 1117–1147, 2002.

L. Ljungqvist and T. J. Sargent. *Recursive macroeconomic theory*. MIT press, 2004.

T. Lux and M. Marchesi. Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719):498–500, 1999.

C. M. Macal and M. J. North. Tutorial on agent-based modeling and simulation. In *Proceedings of the 37th conference on Winter simulation*, pages 2–15. Winter Simulation Conference, 2005.

A. Mandel and H. Gintis. Decentralized pricing and the strategic stability of walrasian general equilibrium. 2014.

L. Meisser and F. Kreuser. An agent-based simulation of the stolper-samuelson effect. In *Proc. of the 21st Int. Conf. on Computing in Economics and Finance (CEF 2015)*, 2015.

P. Milgrom and J. Roberts. *Economics, organization and management*. Prentice-Hall, 1994.

T. Negishi. The stability of a competitive economy: A survey article. *Econometrica: Journal of the Econometric Society*, pages 635–669, 1962.

H. V. D. Parunak, R. Savit, and R. L. Riolo. Agent-based modeling vs. equation-based modeling: A case study and users' guide. In *Multi-agent systems and agent-based simulation*, pages 10–25. Springer, 1998.

J. Poterba. The impact of population aging on financial markets. Technical report, National Bureau of Economic Research, 2004.

E. C. Prescott and R. Mehra. Recursive competitive equilibrium: The case of homogeneous households. *Econometrica*, 48(6):1365–1379, 1980.

T. Rader. Pairwise optimality and non-competitive behavior. *Papers in Quantitative Economics*, 1, 1968.

E. Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999.

J. W. Reeves. Code as design: Three essays. *Developer. The Independent Magazine for Software Professionals*, pages 1–24, 2005.

L. Riccetti, A. Russo, and M. Gallegati. An agent based decentralized matching macroeconomic model. *Journal of Economic Interaction and Coordination*, pages 1–28, 2014.

J. Rouchier. The interest of having loyal buyers in a perishable market. *Computational Economics*, 41(2): 151–170, 2013.

J. Rust, R. Palmer, and J. H. Miller. Behaviour of trading automata in a computerized double auction market. Santa Fe Institute, 1992.

I. Salle, M. Yıldızoğlu, and M.-A. Sénégas. Inflation targeting in a learning economy: An abm perspective. *Economic Modelling*, 34:114–128, 2013.

E. Samanidou, E. Zschischang, D. Stauffer, and T. Lux. Agent-based models of financial markets. *Reports on Progress in Physics*, 70(3):409, 2007.

P. A. Samuelson. An exact consumption-loan model of interest with or without the social contrivance of money. *The journal of political economy*, pages 467–482, 1958.

T. J. Sargent. Bounded rationality in macroeconomics: The arne ryde memorial lectures. *OUP Catalogue*, 1993.

H. Scarf and T. Hansen. The computation of economic equilibria. *New Haven*, 1973.

P. Seppecher. A java agent-based macroeconomic laboratory. 2012.

J. Shore. Fail fast [software debugging]. *Software, IEEE*, 21(5):21–25, 2004.

H. A. Simon. Rational decision making in business organizations. *The American economic review*, pages 493–513, 1979.

C. A. Sims. Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, pages 1–48, 1980.

C. A. Sims. A simple model for study of the determination of the price level and the interaction of monetary and fiscal policy. *Economic theory*, 4(3):381–399, 1994.

H. Sonnenschein. Do walras' identity and continuity characterize the class of community excess demand functions? *Journal of economic theory*, 6(4):345–354, 1973.

W. F. Stolper and P. A. Samuelson. Protection and real wages. *The Review of Economic Studies*, 9(1):58–73, 1941.

L. Tesfatsion. Agent-based computational economics: A constructive approach to economic theory. *Handbook of computational economics*, 2:831–880, 2006.

S. Thurner, J. D. Farmer, and J. Geanakoplos. Leverage causes fat tails and clustered volatility. *Quantitative Finance*, 12(5):695–707, 2012.

H. Uzawa. Walras' tatonnement in the theory of exchange. *The Review of Economic Studies*, pages 182–194, 1960.

U. Wolffgang. A multi-agent non-stochastic economic simulator. In *Proc. of the 21st Int. Conf. on Computing in Economics and Finance (CEF 2015)*, 2015.

S. Wolfram. *A new kind of science*. Wolfram Media Inc., Champaign, Ilinois, US, United States, 2002.

R. Wright. Search, evolution, and money. *Journal of Economic Dynamics and Control*, 19(1):181–206, 1995.